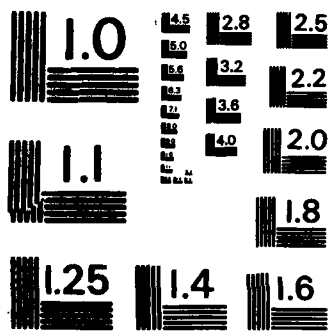
AD-A166 108 IMPLEMENTATION OF THERMAL MODELS AND ALGORITHMS AT FIELD COMMAND DNA (DFF (U) SCIENCE AND ENGINEERING ASSOCIATES INC SALEM MA B S CHAMBERS 12 DEC 84 1/

UNCLASSIFIED SEA-84-4 DNA-TR-85-4 DNA001-83-C-0257 F/G 18/8 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A166 108

(12)

DNA-TR-85-4

IMPLEMENTATION OF THERMAL MODELS AND ALGORITHMS AT FIELD COMMAND, DNA

Burton S. Chambers III
Science and Engineering Associates, Inc.
76 Lafayette Street
Salem, MA 01970

12 December 1984

Technical Report

CONTRACT No. DNA 001-83-C-0257

Approved for public release;
distribution is unlimited.

THIS WORK WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B345083466 G55AMXGR00015 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, DC 20305-1000

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A166108

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188
Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY N/A since Unclassified	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A since Unclassified	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) SEA 84-4	5. MONITORING ORGANIZATION REPORT NUMBER(S) DNA-TR-85-4
6a. NAME OF PERFORMING ORGANIZATION Science and Engineering Associates, Inc. 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Director Defense Nuclear Agency
6c. ADDRESS (City, State, and ZIP Code) 76 Lafayette Street Salem, MA 01970	7b. ADDRESS (City, State, and ZIP Code) Washington, DC 20305-1000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DNA 001-83-C-0257
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62715H	G55AMXG	R	DH006998

11. TITLE (Include Security Classification)
IMPLEMENTATION OF THERMAL MODELS AND ALGORITHMS AT FIELD COMMAND, DNA

12. PERSONAL AUTHOR(S)
Chambers, Burton S. III

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Report	FROM 830525 TO 840331	841212	122

16. SUPPLEMENTARY NOTATION
This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B345083466 G55AMXGR00015 H2590D.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
14	2	
20	13	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Modeling of Simulated Environments
Thermal Radiation Simulators (TRS)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes an implementation of a model on a machine at Field Command, DNA. This model estimates the radiated thermal environment generated by aluminum-LOX thermal radiation simulators. This work was accomplished so that FCDNA could estimate TRS environments at their facility.

See p 10

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Betty L. Fox	22b. TELEPHONE (Include Area Code) (202) 325-7042 22c. OFFICE SYMBOL DNA/STTI

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

SUMMARY

Science and Engineering Associates, Inc. (SEA) has performed the following tasks during this effort.

SEA purchased hardware and licensed software needed to implement DNA ALFGEE models and algorithms at Field Command DNA as specified in Tasks 2 through 5. The hardware and software were identified in Section 3 of this report. The hardware and software was tested at SEA during the warranty periods and the hardware's and software's integrity were shown to be in working order, with one exception, which was discovered after the warranty period had lapsed. One of the disks developed intermittant write errors that can sometimes erase the directory. SEA has purchased a replacement disk drive and has had the other repaired. The repaired unit is intended to be used as a spare.

SEA implemented DNA ALFGEE algorithms and models on the hardware purchased in Task I above for the language and operating system licensed in Task I above and described in Section 3. The resultant implementation calculates the received thermal radiation from an ensemble (no more than 50) of DNA Al-LOX TRS nozzles, that are vertically oriented as currently used at the DNA temporary site at Kirtland AFB, New Mexico.

SEA revised the implementation of the DNA ALFGEE algorithms and models developed in Task 2 above to include an option for use of English units (feet) when specifying nozzle or detector locations. Flux and other variables were not affected by this change.

Acceptance tests were selected by SEA, with DNA approval, and were run at SEA before Task 5 was initiated. The implementation performed as intended.

SEA installed the implementation of the DNA ALFGEE algorithms and

models developed in Task 3 above at Field Command DNA and trained the TRS Technical Director of FCDNA how to use the implementation developed in Tasks 2 and 3 above. Acceptance by Field Command of the FC-TEE occurred within two days of installation. Replication of results in Task 4 constituted acceptance. SEA also installed a second system at Headquarters, DNA.

SEA annotated a listing of the source code, that describes all the algorithms and models relevant to representing the thermal radiated environment from the DNA Al-LOX TRS, used to effect the implementation developed in Tasks 2 and 3 above. This annotated listing was provided to DNA Headquarters and Field Command DNA, and has been included as Section 6 of this report.

PREFACE

This effort was performed by Science and Engineering Associates, Inc. (SEA) for Defense Nuclear Agency (DNA) under DNA Contract Number 001-83-C-0257. The SEA principal investigator was Mr. Burton S. Chambers III. The DNA Contracting Technical Manager (CTM) was LTC Robert Flory, USA. The work involved implementing computer models and algorithms on two machines purchased by SEA for the government. These machines were delivered to Capt. Edward Raska, USAF of Field Command, DNA (FCDNA) and LTC Robert Flory at Headquarters, DNA.

The delivered implementations were for the Apple Pascal Operating System, Version 1.1 on the Apple IIe. Other software was purchased for the government. The software delivered under this effort is not generally available, and may only be used on the two machines delivered under this effort.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

Defense Nuclear Agency (DNA) has been supporting efforts for the past seven years whereby thermal radiation from nuclear weapons can be simulated. These efforts have produced both large area thermal simulators with moderate fidelity and high fidelity smaller laboratory ones. DNA has also been supporting the development of models and algorithms, that are designed to be used to estimate the radiated environments from these simulators, for the past five years. see #1473

I-1 HISTORY OF DNA THERMAL RADIATION SIMULATORS (TRS).

The first significant large area irradiation system was the so-called bag system developed by Science Applications, Inc. (SAI) under contract to DNA. The bag system consisted of an ensemble of mylar bags, each a hemispherically capped cylinder about 6 meters high and 1.5 meters in diameter. Each bag was filled with gaseous oxygen into which was sprayed aluminum powder and then ignited. That system was fielded during the late seventies at a temporary site used by DNA at Kirtland AFB where various targets of military significance were exposed to thermal-only environments. These tests allowed a qualitative look at the effects induced from large area irradiation of large targets. Effects such as panel buckling were observed that were never seen on smaller coupon size samples. The data from these types of tests are extremely useful for validating large complex structural analysis computer models and tools. Eventually, one ensemble of these TRS bags was fielded on a DNA high explosive (HE) event, MISERS BLUFF.

A significant advancement in simulators was achieved when DNA had SAI develop the Al-LOX TRS flame system. This system successfully avoids the undesirable features of the earlier bag systems. Four ensembles of the flame simulators were fielded on HE event, MILL RACE. Each ensemble consisted of

either four or eight nozzles regularily spaced at 3.5, 7, or 10.5 feet apart. Each nozzle rapidly mixes aluminum powder (fluidized in a nitrogen bed) with liquid oxygen. This mixture is sprayed upward into the air and is ignited with a propane burner placed just above the nozzle. This produces a bright flame, which can burn until the materials being injected are exhausted. Due to the time taken for the burning of the aluminum particles, and the initial momentum associated with the spray, as well as buoyancy forces, the flame from each nozzle typically reaches a significant height of three to eight meters. The radiating part of each flame appears as an inverted cone of height about 4.4 meters and diameter of 2.7 meters at the cone's "base". Arranging the nozzles equally spaced along a line parallel to the ground can literally generate a wall of flame.

Another, equally important, advancement in simulators was achieved when DNA had SAI design and build the Phase-I Flashlamp Thermal Radiation Simulator (FTS-I), which has been completed. The Phase I-system readily achieved its design goals. The Phase II-system is presently undergoing checkout.

1-2 USE OF MODELS AND ALGORITHMS FOR AL-LOX TRS.

The models and algorithms describing the radiated environment from Al-LOX TRS have been well exercised. So much so, in fact, that when predictions are needed to position targets and flux calorimeters, Field Command, DNA (FCDNA) has frequently interpolated (and sometimes extrapolated) to other conditions based on existing calculations.

Although the TRS models developed for DNA thus far do a reasonably good job in replicating the data taken in the field for the conditions close to those upon which the models were based, the practice of interpolating on results only adds additional uncertainty. Since the models accurately predict the environments, which should not be surprising since the model parameters are determined by calibration to the data, they should be used for all predictions. However, it has sometimes been inconvenient for FCDNA to have

SEA do these calculations.

Therefore, it was decided to implement the models and algorithms at FCDNA on a particular microcomputer, thus providing FCDNA with its own capability for directly performing these calculations.

SECTION 2

BACKGROUND

This section presents a very brief description of the models used for DNA TRS thermal environment estimation. A more complete description can be found in Section 5. For convenience, all of the various models and algorithms discussed here and used to predict the thermal environments, are called ALFGEE, an acroymn for Aluminum-LOX Field Generated Environment Estimator. Basically, these models are reasonably simple numerical simulations of each flame. They are used to calculate the radiance at desired positions in the field at an imaginary detector oriented in some specified direction.

2-1 INTRODUCTION TO ALFGEE.

The ALFGEE models and algorithms have been implemented as a computer program for FCDNA, which is called the Field Command Thermal Environment Estimator (FC-TEE). This program computes the expected radiance at a field point placed in front of a set of TRS flames.

FC-TEE is based on a program written by a SEA staff member on another microcomputer. The original program has historically been used to obtain predictions for TRS sources that are usually spaced regularily on the ground. In both versions, the analyst interactively specifies the positions of each TRS nozzle and each field point at which the desired environment is to be calculated. The calculations, or predictions, can be printed or written to a file for subsequent processing.

The ALFGEE models can be used to determine model parameters by comparing with experimental data. Usually, only a subset of the existing data base is chosen for comparison. Some measurements are excluded because the data are known to be suspect, for example, those made with defective calorimeters. Others are excluded because they aren't representative. For example, when the

aluminum and LOX flow rates are set to be much different than the standard practice.

The approach to setting model parameters is to choose ones that identically yield a zero residual sum when the data base is compared with predictions based on the chosen parameter set. Each set of parameters that satisfy this condition defines a candidate model. The sum of the variances for each of these different candidate models are then compared. The candidate model with the lowest variance sum is considered the best model for the selected data base. However, it is wise to inspect individual comparisons to be sure differences are random in nature, rather than biased. This approach has been found to be a satisfactory procedure for determining model parameters.

The FC-TEE implementation of ALFGEE is based on model improvements made until April 1983. This version more accurately estimates the radiated environment from those DNA Al-LOX TRS units built and fielded on MISTY CASTLE, Event MILL RACE, than any other earlier model version. The most significant improvement consisted of treating each flame as an inverted cone, rather than as a right cylinder. Account of flame obscuration was done in a manner similar to the approach taken for the cylindrical representation. An additional improvement was the account of the change in view angle for the obscured flames. Finally, this version was calibrated. Available data, taken prior, during and subsequent to MILL RACE, was exercised with the newly developed model and compared with the experimentally measured values.

The time necessary to perform calculations with the new models has substantially increased over what used to be the case. However, use of a dedicated processor mitigates most problems associated with slower responsiveness. In addition, a feature has been added that allows the user to avoid the obscuration calculation. This enables must faster calculations but sometimes at the expense of accuracy.

2-2 ALFGEE DATA BASES.

Various data bases have been developed from the experimental measurements recorded during the past two years of TRS firings. Each data base represents a subset of all available TRS data and potentially can be used to calibrate any ALFGEE model. Depending upon the success achieved in replicating the experimental data, some of these various data bases can provide an indication of the overall variability during the actual experiments. Whether the variations are caused by experimental error or instead by shot-to-shot variations is not easily deduced from such an exercise.

Each data base has been chosen as a statistically meaningful collection representing either: (1) a particular type of event, such as all multi-nozzle events where the nozzles are separated by 7 feet, or (2) a subset of the 3.5 feet separation data base, where certain suspicious measurements have been removed. The entire 3.5 feet data base also exists to document what had been excluded in the other subsets. The most suspicious data were removed from earlier events in an attempt to avoid skewing the data with experiments performed with different reagent flow rates.

SECTION 3

USE OF THE IMPLEMENTATION

The microcomputer, which was installed at FCDNA, its software, and the SEA implementation of the DNA algorithms and models describing Al-LOX TRS radiated thermal environments is referred to as FC-TEE.

SEA chose the Apple IIe (Apple is a registered trademark of Apple Computer, Inc.) as the main component in the FC-TEE hardware. The selection criteria are listed in Section 3-1. SEA installed the FC-TEE hardware listed in Section 3-2. The FC-TEE software is indicated in Section 3-3.

3-1 SELECTION CRITERIA FOR THE FC-TEE SYSTEM.

The following selection criteria were adopted to choose the Apple IIe.

a. The microcomputer should have a minimum of 64K bytes, where K is defined as 1,024.

b. The operating system should be equivalent to that known as Apple Pascal 1.1, which is a derivative of the University of California San Diego (UCSD) Pascal Version II.1.

c. The high level language should have identical implementation features to Apple Pascal 1.1.

d. The microcomputer should support an 80 column screen format to allow efficient program modification.

e. The microcomputer should be able to run the Apple Business Graphics program, which will allow Field Command to generate plots of the

data.

f. The hardware should include a printer with graphics capability with an interface that will work with Apple Business Graphics with no additional programming necessary.

g. The microcomputer could directly read the diskettes containing the source image of the Apple II Plus implementation available to the SEA principal investigator. The format is Apple Pascal TEXT file.

h. The entire system could run with minor modification the Apple II Plus implementation available to the proposed principal investigator.

3-2 CONFIGURATION OF THE FC-TEE HARDWARE.

The FC-TEE hardware was configured by SEA with the following components and the indicated slot allocation.

Quantity	Item (Model)	Manufacturer	Slot No.
1	Apple II e (with Revision B or later motherboard, Revision A motherboard is NOT acceptable)	Apple Computer	N/A
1	Apple Disk II (with controller)	Apple Computer	6
1	Apple Disk II	Apple Computer	6
1	FX-80 Printer	Epson America	N/A
1	Grappler+ Printer Interface or equivalent	Orange Micro	1

1	Apple Extended 80-Column Text Card (includes 64K memory)	Apple Computer	Aux

3-3 SOFTWARE TO RUN FC-TEE.

The FC-TEE software has been purchased or leased by SEA for the government and includes the following.

a. Apple Pascal 1.1. This includes the Operating System, an Editor, a Filer, a Compiler for Apple Pascal, a Linker, a disk Formatter, and a run-time Library containing the following necessary code to be able to run the implementation: transcendental functions, Pascal I/O routines, keyboard sensing routine. This system was purchased by SEA for the government. The license to use same should be executed when the system is received, but before use.

b. Apple Business Graphics. This includes routines that allow generation of graphics from data contained in Apple Pascal TEXT files, or specified at the keyboard. This program was purchased by SEA for the government. The license to use same should be executed when the system is received, but before use.

c. Real Number I/O Routines. These include routines for allowing user input of data without possibility of run-time errors associated with keystroke mistakes, and for trapping all mathematical errors that would cause run-time errors, e.g. arithematic underflow. All data are input through these routines in the current implementation. These routines were licensed by SEA and may not be further distributed. They have been incorporated in the package.

3-4 SEA-PROVIDED IMPLEMENTATION.

The algorithms and models developed by SEA were implemented in executable form for FC-TEE under the UCSD Pascal (trademark of the University of California Regents) operating system, Version II.1, Subset: Apple Pascal (a registered trademark of Apple Computer, Inc.) Version 1.1 released in late 1980 (hereinafter referred to as the Operating System).

Since the vendor of the Operating System requires an end-user license (with a one-time fee at purchase time) to be executed for use on each microcomputer-owner combination (or so it appears), SEA purchased two copies of the Operating System specifically for use only with the FC-TEE machine by the FCDNA TRS Technical Director and his counterpart at DNA. SEA did not use these copies, but delivered them to FCDNA and DNA with the FC-TEE systems. SEA licensed its own copy of the Operating System for its use during development, thus avoiding any potential problems associated with cross-licensing.

3-5 DELIVERABLES.

SEA delivered the following items under this contract:

a. One microcomputer, its software and the SEA implementation as described above to DNA Field Command.

b. A second microcomputer, its software and the SEA implementation as described above to Headquarters DNA.

c. A brief Users' Manual. This manual briefly describes how to use the Program and explains the format for all Input and Output to the Program, which will include any intermediate TEXT files.

d. A brief letter final report with a brief description of SEA's recommendation of how the hardware configuration of the FC-TEE could be enhanced, if this is appropriate.

e. An annotated listing of the physics models and numerical algorithms used in the implementation.

Items c, d and e have been combined into this one report.

SEA applied its best efforts in applying its knowledge to designing and implementing the FC-TEE. However, due to the inherent limitations of equipment such as those purchased for this effort, SEA shall have no liability or responsibility to DNA or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by the Program, including, but not limited to, any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of this Program.

SECTION 4

INTERACTIVE INPUT

This section presents the set of screens, ie. CRT images, that FC-TEE produces interactively for the user. They were also described in detail in a draft narrated videotape (1/2" VHS format) provided to DNA and FCDNA.

4-1 COMMAND FILE.

The command file that produces the set of screens shown in Section 4-2 follows. A command file simply contains a record of the sequence of keyboard strokes a user has made when the command file was written and is saved by the operating system as a TEXT file. The carriage return character (ASCII 13 in base 10) is shown below as a new line. The first character, which is a percent symbol in the example below, tells the system what the terminator is. The convention is that a double terminator symbol terminates the executing TEXT file. The system is left in whatever state results with the keyboard enabled for further data.

```
%xxlox/flame
14
BC27.4
0313y-10.5
00414.5
00513
210
31
060725mine
085x3y090y%%%%
```

4-2 EXAMPLE SET OF SCREENS FROM USING EXAMPLE COMMAND FILE.

Some similar screens that result from executing the command file in Section 4-1 have been removed and the rest are shown and described in this section. In any case, the keystrokes that lead to the particular screen are shown. The screens shown below were also described in detail in a draft narrated videotape (1/2" VHS format) provided to DNA and FCDNA. However, the case shown in the videotape does not necessarily duplicate the one shown here.

The first screen the user sees when the program is loaded is shown as Figure 1. The first line is a header showing: the version of Apple Pascal (1.1), the title of this implementation, "FC-TEE", and the version of the implementation, BSC(17.8). Next comes a line that indicates which screen is being viewed. This particular screen is the LEVEL i menu, which is entitled "GENERAL options".

Following the "level" line (ignoring the ones with dashes) are 14 lines (1 through e), each with an option number, a description, and a value column. On this particular screen only 6 lines (lines 1, a, b, c, d, and e) contain values.

The first four lines (numbered 1 through 4) are used to set data relevant to the flames. For example, selection of the 1st line, entitled "number of TRS nozzles in array", allows you to specify how many nozzles you want; its initial value is one. Selection of one of the rest of these first four lines, takes you to a menu where you can change other TRS flame parameters. Selection of the 2nd line entitled "TRS nozzle characteristics", for example, rewrites the screen and shows options which can be used to set the TRS model parameters. Similarily, the 3rd line, entitled "TRS nozzle position selector", takes you a menu where you can set the nozzle positions. Selection of the 4th line entitled "TRS nozzle strength selector" allows you to modify individual nozzle strengths.

The next two lines (numbered 5 and 6) are used to set data relevant to the detectors. For example, selection of the 5th line, entitled "Detector

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 1 menu:       GENERAL options
----------------------------------------
 #          Description           Value
---    -------------------     ---------
 1 Number of TRS nozzles in array  1
 2 TRS nozzle characteristics
 3 TRS nozzle position selector
 4 TRS nozzle strength selector
 5 Detector position
 6 Detector orientation
 7 Output format specifications
 8 Specify loop order
 9 RUN the case
 a (toggle) Ignore keypress ?    no
 b (toggle) Ignore obscuration ? no
 c (toggle) English units ?      no
 d (toggle) Use AB Graphics ?    no
 e (toggle) Path Input ?         no
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 1. Original Default Screen.

position" allows placement of the detectors. The 6th line allows specification of the detector's orientation.

The next two lines (numbered 7 and 8) affect where the results are sent and what order they are calculated. These lines are entitled "Output format specifications", and "Specify loop order".

The 9th line, entitled "RUN the case", is selected when you have specified all the data needed for the case you want calculated and you are ready to have the calculation performed (RUN).

Finally the last five lines (numbered a through e) are toggles. Selection of any one will change the option to the opposite of what was originally displayed. All options have a default value of "no". The first toggle is "Ignore keypress ?". If its value is no, you can interrupt a calculation by pressing the escape key. If on the other hand you want the calculation to run without keyboard interruption, you select option "a", and its value will change to "YES". Similarily, option b is toggled to ignore obscuration, option c for English units, option d to use Apple Business (AB) Graphics, and e for Path Input.

Figure 2 shows the same screen as Figure I, except that keystrokes "I,4,cr,B,C" have been entered. The convention for showing keystrokes in this section is to enclose them with quotes and separate them with commas. All strings are entered as a series of characters. Note that in all cases a carriage return (or simply RETURN) is represented by the two characters "cr". The RETURN key is pressed. Since it generates a non-printing character, we artificially represent it in this document by this two character sequence.

Figure 3 shows a new screen that results after keystrokes "2,7,.4,cr" have been entered. While this screen is visible, you can change the flame's model parameters. Each flame in the ensemble must be of the same type and hence, these parameters only need to be changed once. Since the previous screen was set for English units, the spatial parameters are given in feet.

```
Pascal (1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 1 menu:        GENERAL options
----------------------------------------
 #          Description            Value
---    -------------------     ---------
 1 number of TRS nozzles in array   4
 2 TRS nozzle characteristics
 3 TRS nozzle position selector
 4 TRS nozzle strength selector
 5 Detector position
 6 Detector orientation
 7 Output format specifications
 8 Specify loop order
 9 RUN the case
 a (toggle) Ignore keypress ?     no
 b (toggle) Ignore obscuration ? YES
 c (toggle) English units ?      YES
 d (toggle) Use AB Graphics ?     no
 e (toggle) Path Input ?          no
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 2. Screen after keystrokes "1,4,cr,B,C".

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 2 menu:      TRS NOZZLE specs
----------------------------------------
 #         Description            Value
---   -------------------      ---------
 1 RELATIVE STRENGTH (-)   =      1.0000
 2 ABS.strength (cal/sec) =   1.53372E7
 3 Flame Diameter (ft.)    =        9.00
 4 Flame Height    (ft.)   =       14.40
 5 Hgt of luminosity(rel) =        1.00
 6 Strength @ nozzle(rel) =       25.00
 7 Ground albedo   (0...1) =        0.40
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 3. Screen after keystrokes "2,7,.4,cr".

The 1st line is the standard flame's strength relative to its absolute strength. Its default value is 1.0000. The 2nd line is the standard flame's absolute strength and its default value is 1.53372E7. The 3rd line is the standard flame's diameter and its default value is 9.00. The 4th line is flame height and its default value is 14.40. The 5th line is the height of luminosity relative to the flame's height and its default value is 1.00. The 6th line is the flame's strength at the nozzle, relative to a value of 1 at the flame's height, and its default value is 25.00. The 7th line is ground albedo and its default value is 0.20. In this case we have changed its value to 0.40.

Figure 4 results after keystroke "0,3". This screen is shown to allow you to select which set of nozzles contains the one you wish to modify. Since ensembles can be larger than can be contained on one screen, this more general approach was taken. Figure 5 is the screen generated after keystrokes "1,3,y,-10.5,cr". Here you can change the coordinates of the nozzles. Each nozzle is always oriented upward, along the z-coordinate in its positive direction. The ground, which is a reflector, is at z = 0.

Figure 6 results after keystroke "0". Figure 7 results after keystrokes "0,4,1,4,.5,cr". While this latter screen is visible you can change the strength of each nozzle relative to the standard one.

Figure 8 results after keystrokes "0,0,5". While this screen is visible, you may change the locations where the detector is placed. Variables ending in a zero are initial positions, those ending with the letter f are final positions, and those beginning with the letter d are the increments (or spacing).

Figure 9 results after keystrokes "1,3,cr" and shows how to change the initial x position to 3.00. Figure 10 shows the screen after keystrokes "2,10,cr". This changes the final x position to 10.00. Figure 11 shows the screen after keystrokes "3,1,cr". Note that this changes the number of cases to 8, since dx was set to 1 and there are 8 cases between 3 ft. and 10 ft. of 1 ft. spacing.

```
Pascal(1.1)         FC-TEE       BSC(17.8)

TRS: LEVEL 2 menu:            NOZZLE lists
----------------------------------------
 #          Description            Value
---   -------------------    ---------
 1       nozzles    ( 1 -  4)        *
----------------------------------------
Option #: [0 returns]     Which one ?
```

Figure 4. Screen after keystroke "0,3".

```
Pascal(1.1)         FC-TEE       BSC(17.8)

TRS: LEVEL 3 menu:  TRS NOZZLE locations
----------------------------------------
# m       x (ft)       y (ft)       z (ft)
- --  ----------  ----------  ----------
1 1      0.00000    -4.50000     0.00000
2 2      0.00000     4.50000     0.00000
3 3      0.00000   -10.5000      0.00000
4 4      0.00000    13.5000      0.00000
----------------------------------------
Option #: [0 returns]     Which one ?
```

Figure 5. Screen after keystrokes "1,3,y,-10.5,cr".

```
Pascal(1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 2 menu:           NOZZLE lists
----------------------------------------
 #          Description            Value
---   ------------------     ---------
 1        nozzles    ( 1 -  4)       *
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 6. Screen after keystroke "0".

```
Pascal(1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 3 menu:  TRS NOZZLE strengths
----------------------------------------
#  Mod   Strength ( - )
-  ---   --------------  -------------
1   1       1.00000
2   2       1.00000
3   3       1.00000
4   4       0.50000
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 7. Screen after keystrokes "0,4,1,4,.5,cr".

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 2:   DETECTOR position in ft.
----------------------------------------
 #          Description             Value
---    -------------------      ---------
 1      xdist0: smallest x =        3.28
 2      xdistf: greatest x =        3.28
 3       dx: spacing in x  =        0.00

 4      ydist0: smallest y =        0.00
 5      ydistf: greatest y =        0.00
 6       dy: spacing in y  =        0.00

 7      zdist0: smallest z =        4.92
 8      zdistf: greatest z =        4.92
 9       dz: spacing in z  =        0.00

Number of cases is  =         1.0
nx:    1    ny:    1    nz:    1
na:    1    ne:    1
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 8. Screen after keystrokes "0,0,5".

```
Pascal(1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 2:   DETECTOR position in ft.
----------------------------------------
 #          Description            Value
---    -------------------     ---------
 1      xdist0: smallest x =        3.00
 2      xdistf: greatest x =        3.28
 3       dx: spacing in x  =        0.00

 4      ydist0: smallest y =        0.00
 5      ydistf: greatest y =        0.00
 6       dy: spacing in y  =        0.00

 7      zdist0: smallest z =        4.92
 8      zdistf: greatest z =        4.92
 9       dz: spacing in z  =        0.00

Number of cases is  =         1.0
nx:   1   ny:   1   nz:   1
na:   1   ne:   1
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 9. Screen after keystrokes "1,3,cr".

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 2:   DETECTOR position in ft.
----------------------------------------
 #          Description             Value
---     -------------------     ---------
 1      xdist0: smallest x =         3.00
 2      xdistf: greatest x =        10.00
 3       dx: spacing in x  =         0.00

 4      ydist0: smallest y =         0.00
 5      ydistf: greatest y =         0.00
 6       dy: spacing in y  =         0.00

 7      zdist0: smallest z =         4.92
 8      zdistf: greatest z =         4.92
 9       dz: spacing in z  =         0.00

Number of cases is  =          1.0
nx:    1    ny:    1    nz:    1
na:    1    ne:    1
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 10. Screen after keystrokes "2,10,cr".

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 2:   DETECTOR position in ft.
----------------------------------------
 #          Description             Value
---    -------------------     ---------
 1     xdist0: smallest x =        3.00
 2     xdistf: greatest x =       10.00
 3      dx: spacing in x  =        1.00

 4     ydist0: smallest y =        0.00
 5     ydistf: greatest y =        0.00
 6      dy: spacing in y  =        0.00

 7     zdist0: smallest z =        4.92
 8     zdistf: greatest z =        4.92
 9      dz: spacing in z  =        0.00

Number of cases is   =         8.0
nx:   8   ny:   1   nz:   1
na:   1   ne:   1
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 11. Screen after keystrokes "3,1,cr".

Figure 12 shows a similar screen after keystrokes "0,6". It corresponds to the detector orientation. These parameters are modified in a manner similar to those shown for the x coordinate. Note that both the letters a and t are used interchangeably to represent azimuth.

Figure 13 shows the screen after keystrokes "0,7" and while visible, the user is allowed to change the OUTPUT specs. Pressing 1 makes CONSOLE: receive output. This is the volume name for the CRT screen. Pressing 2 makes PRINTER: receive output if on-line. This is the volume name for the device attached to Slot #1 on the Apple. Normally, a printer is attached there. Pressing 3 makes REMOUT: receive output if on-line. This is the volume name for the device attached to Slot #2 on the Apple. Normally, another printer, a plotter, or a modem is attached there. Pressing 4 makes a DISK file receive it. Selection of this option will lead to questions about which volume and filename will receive the output. The default name of the output device is CONSOLE:. Pressing 5 allows you to change the Apple Business Graphics (ABG) prefix, which defaults to ABG.

Figure 14 shows the same screen as in Figure 13 but after keystrokes "2,5,mine,cr", which change the output device to the PRINTER: and the ABG prefix to "mine".

Figure 15 is shown after you press keystrokes "0,8". This screen shows the default screen for setting the looping order about detector position and orientation. Figure 16 results after keystrokes "5,x,3,y" where we have changed the loop order. The outer-most loop is over the variable that cycles LEAST rapidly and the inner-most loop is over the variable that cycles MOST rapidly. FC-TEE can only allow five unique letters to exist in this table , ie. no two can be the same, therefore FC-TEE will not allow you to leave this menu until this condition is satisfied.

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 2 menu:  DETECTOR orientation
----------------------------------------
 #          Description            Value
---    -------------------     ---------
 1    azimuth0: smallest   =        0.0
 2    azimuth:  greatest   =        0.0
 3    da:azimuth spacing   =        0.0

 4    elevation0:smallest  =        0.0
 5    elevationf:greatest  =        0.0
 6    de:elevation spacing =        0.0

Number of cases is  =        8.0
nx:   8   ny:   1   nz:   1
na:   1   ne:   1
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 12. Screen after keystrokes "0,6".

```
Pascal(1.1)        FC-TEE       BSC(17.8)

TRS: LEVEL 2 menu:           OUTPUT specs
----------------------------------------
 #          Description            Value
---    -------------------     ---------
 1     make CONSOLE: receive output
 2     make PRINTER: receive output
 3     make REMOUT:  receive output
 4     make a DISK file receive it

        outdev is                CONSOLE:

 5     Change ABG prefix
                     ABG prefix is   ABG
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 13. Screen after keystrokes "0,7".

```
Pascal(1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 2 menu:          OUTPUT specs
----------------------------------------
 #         Description           Value
---    -------------------    ---------
 1    make CONSOLE: receive output
 2    make PRINTER: receive output
 3    make REMOUT:  receive output
 4    make a DISK file receive it

       outdev is                PRINTER:

 5    Change ABG prefix
                      ABG prefix is  mine
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 14. Screen after keystrokes "2,5,mine,cr".

```
Pascal(1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 2 menu:           LOOP control
----------------------------------------
 #          Description             Value
---    -------------------     ---------
 1          outermost loop is over  E
 2          next                    T
 3          next                    X
 4          next                    Z
 5          innermost loop is over  Y

x,y,z,t,e should each only be used once!
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 15. Screen after keystrokes "0,8".

```
Pascal(1.1)        FC-TEE      BSC(17.8)

TRS: LEVEL 2 menu:          LOOP control
----------------------------------------
 #          Description            Value
---    -------------------    ---------
 1          outermost loop is over  E
 2          next                    T
 3          next                    Y
 4          next                    Z
 5          innermost loop is over  X

x,y,z,t,e should each only be used once!
----------------------------------------
Option #: [0 returns]    Which one ?
```

Figure 16. Screen after keystrokes "5,x,3,y".

SECTION 5

EXAMPLE OUTPUT

This section provides some examples of the output that can be obtained with FCTEE. There are two principal formats. The first is designed to be read by a person. The second is intended to be read by the machine, specifically a commercially available program, Apple Business Graphics (ABG), that was purchased for this effort and delivered to FCDNA and DNA. The second format consists of commands to the ABG program customized by FC-TEE.

5-1 FORMAT FOR PEOPLE.

This section provides one FCTEE example of the output for people. It is normally sent to a printer. This example's spatial-dimensions are in feet, however, the results can be produced in centimeters.

Pascal(1.1) FC-TEE Inverted Cone TRS Estimator (NO Obscuration) BSC(17.8)

Flame:	#	X	Y	Z	T	S
	1	0.000	-4.500	0.000	0.00	1.00
	2	0.000	4.500	0.000	0.00	1.00
	3	0.000	-13.500	0.000	0.00	1.00
	4	0.000	13.500	0.000	0.00	1.00

Relative Strength	=	1.0	
Absolute Strength	=	1.53372E7	cal/sec
Flame Diameter	=	9.0	ft.
Flame Height	=	14.4	ft.
Luminosity Limit	=	1.0	(Rel to flame hgt)
"Power" at nozzle	=	25.0	(Rel to power at flame hgt.)

Ground Albedo = 0.2

XDIST (ft.)	YDIST (ft.)	Height (ft.)	Azimuth (deg)	Elevation (deg)	Flux (cal/sqcm/sec)
3.00	0.00	6.00	0.0	0.0	37.24
5.00	0.00	6.00	0.0	0.0	37.51
7.00	0.00	6.00	0.0	0.0	32.09
9.00	0.00	6.00	0.0	0.0	26.55
3.00	5.00	6.00	0.0	0.0	75.82
5.00	5.00	6.00	0.0	0.0	46.22
7.00	5.00	6.00	0.0	0.0	33.76
9.00	5.00	6.00	0.0	0.0	26.53
3.00	10.00	6.00	0.0	0.0	38.76
5.00	10.00	6.00	0.0	0.0	36.36
7.00	10.00	6.00	0.0	0.0	29.87
9.00	10.00	6.00	0.0	0.0	24.13
3.00	0.00	8.00	0.0	0.0	37.97
5.00	0.00	8.00	0.0	0.0	37.99
7.00	0.00	8.00	0.0	0.0	32.28
9.00	0.00	8.00	0.0	0.0	26.56
3.00	5.00	8.00	0.0	0.0	78.06
5.00	5.00	8.00	0.0	0.0	47.05
7.00	5.00	8.00	0.0	0.0	34.02
9.00	5.00	8.00	0.0	0.0	26.56
3.00	10.00	8.00	0.0	0.0	39.61
5.00	10.00	8.00	0.0	0.0	36.88
7.00	10.00	8.00	0.0	0.0	30.08
9.00	10.00	8.00	0.0	0.0	24.15

3.00	0.00	6.00	0.0	45.0	33.27
5.00	0.00	6.00	0.0	45.0	30.40
7.00	0.00	6.00	0.0	45.0	24.86
9.00	0.00	6.00	0.0	45.0	19.89
3.00	5.00	6.00	0.0	45.0	62.93
5.00	5.00	6.00	0.0	45.0	37.02
7.00	5.00	6.00	0.0	45.0	26.12
9.00	5.00	6.00	0.0	45.0	19.88
3.00	10.00	6.00	0.0	45.0	34.18
5.00	10.00	6.00	0.0	45.0	29.40
7.00	10.00	6.00	0.0	45.0	23.13
9.00	10.00	6.00	0.0	45.0	18.09
3.00	0.00	8.00	0.0	45.0	26.21
5.00	0.00	8.00	0.0	45.0	24.72
7.00	0.00	8.00	0.0	45.0	20.79
9.00	0.00	8.00	0.0	45.0	17.03
3.00	5.00	8.00	0.0	45.0	54.24
5.00	5.00	8.00	0.0	45.0	31.09
7.00	5.00	8.00	0.0	45.0	22.03
9.00	5.00	8.00	0.0	45.0	17.06
3.00	10.00	8.00	0.0	45.0	27.20
5.00	10.00	8.00	0.0	45.0	24.14
7.00	10.00	8.00	0.0	45.0	19.43
9.00	10.00	8.00	0.0	45.0	15.52

5-2 FORMAT FOR APPLE BUSINESS GRAPHICS.

This section provides one FCTEE example of the output for ABG. Although, it can be read by people, it is normally sent to a TEXT file for subsequent use. The spatial units for this example are shown in feet, and like the example in Section 5.1, can be produced in centimeters.

```
CO Pascal(1.1)  FC-TEE Inverted Cone TRS Estimator (NO Obscuration) BSC(17.8)
CO
CO
CO Flame:  #          X         Y         Z         T        S
CO         1        0.000    -4.500     0.000     0.00     1.00
CO         2        0.000     4.500     0.000     0.00     1.00
CO         3        0.000   -13.500     0.000     0.00     1.00
CO         4        0.000    13.500     0.000     0.00     1.00
CO
CO  Relative Strength  =          1.0
CO  Absolute Strength  =   1.53372E7  cal/sec
CO  Flame Diameter     =          9.0  ft.
CO  Flame Height       =         14.4  ft.
CO  Luminosity Limit   =          1.0  (Rel to flame hgt)
CO  "Power" at nozzle  =         25.0  (Rel to power at flame hgt.)
CO  Ground Albedo      =          0.2
CO
EDIT
PREFIX _
_E
COMMENT --------------- start of next case ------
CLEAR
CO   ELEVATION =         0.0 (degrees)
CO       THETA =         0.0 (degrees)
CO       ZDIST =      6.00 (ft)
CO       YDIST =      0.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
```

```
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
      3.00      37.24
      5.00      37.51
      7.00      32.09
      9.00      26.55
_E
SAVE ABG1
COMMENT --------------- start of next case ------
CLEAR
CO   ELEVATION =          0.0 (degrees)
CO        THETA =         0.0 (degrees)
CO        ZDIST =       6.00 (ft)
CO        YDIST =       5.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
      3.00      75.82
      5.00      46.22
      7.00      33.76
      9.00      26.53
_E
SAVE ABG2
COMMENT --------------- start of next case ------
CLEAR
CO   ELEVATION =          0.0 (degrees)
CO        THETA =         0.0 (degrees)
CO        ZDIST =       6.00 (ft)
CO        YDIST =      10.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
      3.00      38.76
      5.00      36.36
      7.00      29.87
```

```
        9.00      24.13
_E
SAVE ABG3
COMMENT -------------- start of next case ------
CLEAR
CO   ELEVATION =        0.0 (degrees)
CO       THETA =        0.0 (degrees)
CO       ZDIST =      8.00 (ft)
CO       YDIST =      0.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
        3.00      37.97
        5.00      37.99
        7.00      32.28
        9.00      26.56
_E
SAVE ABG4
COMMENT -------------- start of next case ------
CLEAR
CO   ELEVATION =        0.0 (degrees)
CO       THETA =        0.0 (degrees)
CO       ZDIST =      8.00 (ft)
CO       YDIST =      5.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
        3.00      78.06
        5.00      47.05
        7.00      34.02
        9.00      26.56
_E
SAVE ABG5
COMMENT -------------- start of next case ------
CLEAR
```

```
CO   ELEVATION =        0.0 (degrees)
CO        THETA =       0.0 (degrees)
CO        ZDIST =     8.00 (ft)
CO        YDIST =    10.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
       3.00      39.61
       5.00      36.88
       7.00      30.08
       9.00      24.15
_E
SAVE ABG6
COMMENT -------------- start of next case ------
CLEAR
CO   ELEVATION =       45.0 (degrees)
CO        THETA =       0.0 (degrees)
CO        ZDIST =     6.00 (ft)
CO        YDIST =     0.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
       3.00      33.27
       5.00      30.40
       7.00      24.86
       9.00      19.89
_E
SAVE ABG7
COMMENT -------------- start of next case ------
CLEAR
CO   ELEVATION =       45.0 (degrees)
CO        THETA =       0.0 (degrees)
CO        ZDIST =     6.00 (ft)
CO        YDIST =     5.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
```

```
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
      3.00     62.93
      5.00     37.02
      7.00     26.12
      9.00     19.88
_E
SAVE ABG8
COMMENT -------------- start of next case ------
CLEAR
CO   ELEVATION =        45.0 (degrees)
CO       THETA =         0.0 (degrees)
CO       ZDIST =       6.00 (ft)
CO       YDIST =      10.00 (ft)
SET HORIZONTAL TITLE "Range (ft)"
SET VERTICAL TITLE "Flux cal/sqcm/sec"
EDIT
      3.00     34.18
      5.00     29.40
      7.00     23.13
      9.00     18.09
_E
SAVE ABG9
```

----- NOTE: REST OF CASES DELETED FOR THIS REPORT -----

5-3 PATH INPUT.

This section introduces another example of FCTEE's output. The example shown here can be read by people, but the ABG format could have been selected for use by the machine. The spatial units are the same as in the earlier examples. This example is different in that the "path input" option has been used to generate it. This latter option allows you to specify a path in space, instead of a volume of regularily spaced points. A path in space is selected by option e when the menu in Figure 1 is on the screen. When the option is "YES", the associated meanings attached to the initial and final positions of the detector change. These variables (six of them) specify two end points of a line segment. With the exception of the coordinate that is the innermost loop coordinate (see Figure 15), the delta-coordinate has no effect on the spacing along the line segment. Only the inner loop coordinate's delta controls the spacing.

Pascal(1.1) FC-TEE Inverted Cone TRS Estimator (NO Obscuration) BSC(17.8)

Flame: #	X	Y	Z	T	S
1	0.000	-4.500	0.000	0.00	1.00
2	0.000	4.500	0.000	0.00	1.00
3	0.000	-13.500	0.000	0.00	1.00
4	0.000	13.500	0.000	0.00	1.00

Relative Strength = 1.0
Absolute Strength = 1.53372E7 cal/sec
Flame Diameter = 9.0 ft.
Flame Height = 14.4 ft.
Luminosity Limit = 1.0 (Rel to flame hgt)
"Power" at nozzle = 25.0 (Rel to power at flame hgt.)
Ground Albedo = 0.2

XDIST (ft.)	YDIST (ft.)	Height (ft.)	Azimuth (deg)	Elevation (deg)	Flux (cal/sqcm/sec)
3.00	0.00	6.00	0.0	0.0	37.24
5.00	3.33	6.67	0.0	0.0	45.89
7.00	6.67	7.33	0.0	0.0	32.72
9.00	10.00	8.00	0.0	0.0	24.15
3.00	0.00	6.00	0.0	45.0	33.27
5.00	3.33	6.67	0.0	45.0	34.60
7.00	6.67	7.33	0.0	45.0	22.57
9.00	10.00	8.00	0.0	45.0	15.52

SECTION 6

ANNOTATED PHYSICS LISTING

This section presents the annotated physics listing. The listings follow. The language used is Apple Pascal version 1.1.

```
{-----------------------------------------------------------------------}
                    MATHTRAP  library unit
{-----------------------------------------------------------------------}

                      USES transcend;

  VAR   matherror: (nomatherror,overflow,underflow,dividebyzero,
                         lognegative,sqrtnegative,argtoobig);

        FUNCTION adder (x , y: REAL): REAL;  {  x + y    }
        FUNCTION suber (x , y: REAL): REAL;  {  x - y    }
        FUNCTION muler (x , y: REAL): REAL;  {  x * y    }
        FUNCTION diver (x , y: REAL): REAL;  {  x / y    }
        FUNCTION exper (x: REAL): REAL;      {  EXP(x)   }
        FUNCTION lner  (x: REAL): REAL;      {  LN (x)   }
        FUNCTION loger (x: REAL): REAL;      {  LOG(x)   }
        FUNCTION sqrter(x: REAL): REAL;      {  SQRT(x)  }
        FUNCTION siner (x: REAL): REAL;      {  SIN(x)   }
        FUNCTION coser (x: REAL): REAL;      {  COS(x)   }
        FUNCTION ataner(x: REAL): REAL;      {  ATAN(x)  }
        FUNCTION truncer (x: REAL): INTEGER; {  TRUNC(x) }
{-----------------------------------------------------------------------}
```

```
{-----------------------------------------------------------------------}
                         GENMATH   library unit
{-----------------------------------------------------------------------}

           USES transcend, {$U MATH.TRAP.CODE} mathtrap;

       FUNCTION asiner (x: REAL):    REAL;  { arc sine        }
       FUNCTION acoser (x: REAL):    REAL;  { arc cosine      }
       FUNCTION taner  (x: REAL):    REAL;  { tangent         }
       FUNCTION xtoy   (x, y: REAL): REAL;  { x to y power    }
{-----------------------------------------------------------------------}
```

```
{04 Nov 83 Documentation for PROGRAM trsradiation}
{$S+} {$V-}

PROGRAM trsradiation;

{---------------------------------------------------------------------------}
{   DOCUMENTATION of Algorithms & Models:  TRS Radiation Program - FLAMES   }
{---------------------------------------------------------------------------}
{ This version uses an inverted cone model for the TRS flames and accounts  }
{ for flame obscuration, change in view angle.                              }
{---------------------------------------------------------------------------}
{                      AUTHOR: Burton S. Chambers, III                      }
{---------------------------------------------------------------------------}

USES
{$U *SYSTEM.LIBRARY}          transcend,
{$U SYSLIB4:XTRS.LIBRARY}     mathtrap,
                              genmath;

CONST pi = 3.141592654; maxnobags = 50;

VAR
     fourpi, degtorad, radtodeg, albedo, hgtacoeff, hgtbcoeff: REAL;
     widthofbag, heightofbag, bestguessbag, bagstrength: REAL;
     xbag, ybag, zbag, tbag, sbag: ARRAY[1..maxnobags] OF REAL;
     nbags: INTEGER;
```

```
      quick: BOOLEAN;

      xdist, ydist, zdist, theta, elev, flu: REAL;
      done: BOOLEAN;
{--------------------------------------------------------------------}
   PROCEDURE readdata;
   {This routine simulates data entry}

   BEGIN
{Eg.}
      xdist := 100.0;  { x-distance to detector in cm.                      }
      ydist :=   0.0;  { y-distance to detector in cm.                      }
      zdist := 600.0;  { z-distance to detector (height above ground) in cm.}
      theta :=   0.0;  { Azimuth in degrees                                 }
      elev  :=   0.0;  { Elevation in degrees                               }
      done := TRUE;
      ybag[1] := 0.0;
{Eg.}
   END;
{--------------------------------------------------------------------}
   PROCEDURE default;
   VAR i,j:INTEGER;

   BEGIN
      fourpi := 4.0 * pi;
```

```
    degtorad := pi / 180; radtodeg := 1.0 / degtorad;
    bagstrength := 1.0; { RELATIVE TO BESTGUESSBAG }
    bestguessbag := 1.53372E7; { CAL/SEC }
    widthofbag := 274.32; { CM } {9'}
    heightofbag := 438.912; { CM } {14.4'}
    albedo := 0.2; { OF GROUND }
    hgtacoeff := 1.0;
    hgtbcoeff := 25.0;
    nbags := 1;
    FOR i := 1 TO maxnobags DO
      BEGIN { DEFAULT TO LINE OF EQUALLY SPACED NOZZLES }
        xbag[i] := 0.0; j := (i+1) DIV 2;
        ybag[i] := j * widthofbag - widthofbag/2;
        zbag[i] := 0.0;
        IF ODD(i) THEN ybag[i] := -ybag[i];
        tbag[i] := 0.0; sbag[i] := 1.0;
      END;
    quick := FALSE;
  END;
{-----------------------------------------------------------------------------}
 FUNCTION cintersection (xl, yl, rl, xn, yn, rn: REAL;
                         VAR d, xi1, yi1, xi2, yi2: REAL): BOOLEAN;

 {INPUT}  { xl   = x coordinate of center of the lth circle (cm.) }
          { yl   = y coordinate of center of the lth circle (cm.) }
          { rl   = radius                  of the lth circle (cm.) }
```

```
        { xn   = x coordinate of center of the nth circle (cm.) }
        { yn   = y coordinate of center of the nth circle (cm.) }
        { rn   = radius                 of the nth circle (cm.) }

{OUTPUT} { d    = distance between centers of the two circles (cm.) }
        { xi1  = x coordinate of 1st intersection point (cm.)      }
        { yi1  = y coordinate of 1st intersection point (cm.)      }
        { xi2  = x coordinate of 2nd intersection point (cm.)      }
        { yi2  = y coordinate of 2nd intersection point (cm.)      }

CONST pi = 3.14159265;
VAR alpha, theta: REAL;

BEGIN
  cintersection := FALSE;
  r1 := ABS(r1); rn := ABS(rn); {make negative radii positive}
  d := sqrter(SQR(xn-x1)+SQR(yn-y1));

  IF (d > 0.0) AND (d <= r1 + rn) THEN {at least one intersection exists}
    BEGIN
      theta := ataner(diver(yn-y1,xn-x1));
      IF xn < x1 THEN {its in quadrant II or III} theta := theta + pi;
      alpha := acoser((SQR(r1) - SQR(d) - SQR(rn))/(2.0*d*rn));
      xi2 := xn + rn * coser(theta+alpha);
      yi2 := yn + rn * siner(theta+alpha);
      xi1 := xn + rn * coser(theta-alpha);
```

```
      yil := yn + rn * siner(theta-alpha);
      cintersection := TRUE;
    END;
  END; {calcintersection}
{--------------------------------------------------------------------------}
  PROCEDURE calculate;

  CONST cangle = 90.0; safety = 3.0;
        mmax = 51; { max set on basis of flame hgt to width data }
        mmin =  5; { min set on guess of flame obscuration effect}

  VAR i,j,k,l,lmax,n,m:INTEGER;
      te,ee,se,ce,tt,st,ct:REAL;
      xj,yj,zj,sj,tj,rg,rg2,mp,si,wl,wlr,nwl,nwlr,zk,r,r2,ca:REAL;
      sip,zip:ARRAY[0..mmax] OF REAL;
      dz:REAL;
      thmax,thmin,thavgi,thavgf: REAL;
      delta_flux: REAL;

      wlbag, wlrbag, rg2bag, rgbag,
      pbag, abag, bbag: ARRAY[1..maxnobags] OF REAL;
      kbag: ARRAY[1..maxnobags] OF INTEGER;
  {--------------------------------------------------------------}
      FUNCTION model (z:REAL):REAL;
      {INPUT} { z = height normalized by flame height                   }
     {OUTPUT} { model = normalized power per unit height * width of flame }
```

```
        { eg. sip[i] := model( (zip[i] - zj)/heightofbag)           }
    VAR p: REAL;

    BEGIN
      IF z <= 1.0 THEN {within main luminous portion of flame}
        BEGIN
          IF hgtbcoeff = 1.0 THEN p := 1.0
          ELSE p := 1.0 + (1.0 - z) * (hgtbcoeff - 1.0)
        END
      ELSE IF z < hgtacoeff THEN {above main portion of flame}
        BEGIN
          IF hgtacoeff <= 1.0 THEN p := 0.0 {shouldn't happen}
          ELSE p := 1.0 + (1.0 - z) / (hgtacoeff - 1.0)
        END
      ELSE p := 0; {Above entire flame}

      IF z < 1.0 THEN {inverted cone} model := p * z * (widthofbag/2.0)
      ELSE {cylinder} model := p * (widthofbag/2.0);
    END; { model }
{---------------------------------------------------------------}
  PROCEDURE selector;
  {an efficient sort on range to detector from flames}
  TYPE  item = RECORD
               key: REAL;
               index: INTEGER;
             END;
```

```
VAR kselect: ARRAY[1..maxnobags] OF item;
    i,j,k:INTEGER; x: item;

BEGIN
  FOR j := 1 TO nbags DO kselect[j].key := rgbag[j];
  FOR j := 1 TO nbags DO kselect[j].index := j;

  FOR i := 1 TO nbags - 1 DO
    BEGIN
      k := i; x := kselect[i];
        FOR j := i + 1 TO nbags DO
          IF kselect[j].key < x.key THEN
            BEGIN
              k := j; x := kselect[j];
            END;
          kselect[k] := kselect[i]; kselect[i] := x;
    END;
  FOR j := 1 TO nbags DO kbag[j] := kselect[j].index;
END;
{---------------------------------------------------------------}
PROCEDURE sortbags;
{a misnomer. Calculates geometry terms prior to sorting them}
VAR w2,radius: REAL; j: INTEGER;

BEGIN
  radius := widthofbag / 2.0;
```

```
  FOR j := 1 TO nbags DO
    BEGIN
      xj := xbag[j]; yj := ybag[j];
      rg2 := SQR(xdist-xj)+SQR(ydist-yj); rg := SQRT(rg2);
      wl := (xdist - xj) * st + (ydist - yj) * ct;
      w2 := (ydist - yj) * st - (xdist - xj) * ct;
      wlr := wl * ce;
      wlbag[j] := wl; wlrbag[j] := wlr;
      rg2bag[j] := rg2; rgbag[j] := rg;
      abag[j] := radtodeg * asiner(radius / rgbag[j]);
      { assumes quad I,IV, ie. abag in -90..+90, and not in flame }
      bbag[j] := radtodeg * acoser(wl / rgbag[j]);
      { quad I,II, ie. bbag in 0..180 }
      IF w2 < 0.0 THEN { flame center to left of normal }
        bbag[j] := -bbag[j];
    END;
 END; { sortbags }
{-----------------------------------------------------------------}
 PROCEDURE modbagstrength (l: INTEGER);
 { modify strength of one bag for this pt }
 VAR j, k, m, n: INTEGER;
     ratio,  satio,  angle, anglemax, anglemin,
     zint,   ztip,   zsource: REAL;
     {dist = distance between circle-centers, xil,etc.=intersections}
     dist, xil,yil,xi2,yi2: REAL;
     radius,
```

```
    tang_dist: REAL;     {distance to tangency}
    intersect: BOOLEAN;
    sr, rr, deltaabag: REAL;

{------------------------------------------------------------}
  PROCEDURE switch (VAR x1,x2: REAL);
  {Switches data in x1 with x2 (and vice versa)}
  VAR t: REAL;

  BEGIN
    t := x1; x1 := x2; x2 := t;
  END;
{------------------------------------------------------------}
  FUNCTION determinant (x1,y1,x2,y2,x3,y3: REAL): REAL;
  {Calculates determinant to determine sign}

  BEGIN
    determinant := x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)
  END;
{------------------------------------------------------------}
  FUNCTION sameside: BOOLEAN;
  {Calculates whether two intersecting points are on the same "side"}
  VAR pos1,pos2: BOOLEAN;

  BEGIN {sameside}
    pos1 :=
```

```
     (determinant(xbag[l],ybag[l],xbag[m],ybag[m],xil,yil) > 0.0);
    pos2 :=
     (determinant(xbag[l],ybag[l],xbag[m],ybag[m],xdist,ydist) > 0.0);
    sameside := (posl AND pos2);
  END; {sameside}
{-----------------------------------------------------------}
  PROCEDURE inmod;
  VAR interldist, inter2dist, betal, beta2: REAL;

  {--------------------------------------------------------}
    FUNCTION calcbeta (x,y,d: REAL): REAL;
    {Beta is relative to detector orientation}
    VAR wl,w2,beta: REAL;

    BEGIN {calcbeta}
      wl := (xdist - x) * st + (ydist - y) * ct;
      w2 := (ydist - y) * st - (xdist - x) * ct;
      beta := radtodeg * acoser(wl / d);
      { quad I,II, ie. bbag in 0..180 }
      IF w2 < 0.0 THEN { intersection to left of normal }
        beta := -beta;
      calcbeta := beta;
    END; {calcbeta}
  {--------------------------------------------------------}
  BEGIN {inmod}
    tang_dist  := sqrter(rg2bag[m]-SQR(sr));
```

```
    inter1dist := sqrter(SQR(xdist-xi1)+SQR(ydist-yi1));
    inter2dist := sqrter(SQR(xdist-xi2)+SQR(ydist-yi2));
    beta1 := calcbeta(xi1,yi1,inter1dist);
    beta2 := calcbeta(xi2,yi2,inter2dist);
    IF inter1dist < tang_dist THEN {use intersection point 1}
      anglemin := bbag[m] - beta1;
    IF inter2dist < tang_dist THEN {use intersection point 2}
      anglemax := beta2 - bbag[m];
  END; {inmod}
{-----------------------------------------------------------}
  PROCEDURE modangles;

  BEGIN {modangles}
    angle := radtodeg * asiner(sr / rgbag[m]);
    anglemax := angle; anglemin := angle;
    IF intersect THEN inmod;
    IF thmin > bbag[m] - anglemin THEN
      IF bbag[m] + anglemax > thmin THEN
        thmin := bbag[m] + anglemax;
    IF thmax < bbag[m] + anglemax THEN
      IF bbag[m] - anglemin < thmax THEN
        thmax := bbag[m] - anglemin;
  END; {modangles}
{-----------------------------------------------------------}
BEGIN { modbagstrength }
  radius := widthofbag / 2.0;
```

```
j := 0;

REPEAT {Find the bag index, j, so that kbag[j] points to 1}
  j := j + 1;
UNTIL (1 = kbag[j]) OR (j = nbags);

ratio := 1.0; ztip := zbag[1];

zsource := ABS(zk); {ground is defined as z=0}
IF zsource < ztip THEN ratio := 0.0
ELSE IF zsource < heightofbag + ztip THEN
  ratio := (zsource - ztip)/heightofbag; {truncated inverted cone}

rr := ratio * radius; {for speed}
deltaabag := radtodeg * asiner(rr / rgbag[1]);
thmin  := bbag[1] - deltaabag;
thmax  := bbag[1] + deltaabag;
thavgi := bbag[1];             {0.5 * (thmin + thmax)}

FOR n := 1 TO nbags DO {over all bags}
{Those closer or intersecting with the source bag can obscure}
  BEGIN
    m := kbag[n];
    anglemin := 0.0; anglemax := 0.0; {for neat debug prints}
    IF 1 <> m THEN {check for obscuration}
      BEGIN
```

```
        satio := 1.0; ztip := zbag[m];
        zint := ABS(zdist + (zk-zdist) * (rgbag[m] / rgbag[l]));

        IF zint < ztip THEN satio := 0.0
        ELSE IF zint < heightofbag + ztip THEN
          satio := (zint - ztip)/heightofbag;
        sr := satio * radius; {for speed}

        intersect := cintersection(
                       xbag[l],ybag[l],rr, {source}
                       xbag[m],ybag[m],sr, {obscuror}
                       dist,xil,yil,xi2,yi2);

        IF (intersect AND (NOT sameside)) THEN
                                      {switch intersection points}
          BEGIN
            switch(xil,xi2); switch(yil,yi2);
          END;

        {Note: The next test must definitely be on n & j:  BSC3}
        IF n <= j THEN {the possibly obscuring flame is closer }
          modangles
        ELSE {it is farther away} IF intersect THEN modangles;

      END;
  END;
```

```
      thavgf := 0.5 * (thmin + thmax);
      IF (thmax > thmin) AND (ratio > 0.0) THEN
        pbag[l] := sbag[l] * (thmax - thmin) / (deltaabag + deltaabag)
      ELSE pbag[l] := 0.0;
    END; { modbagstrength }
{-------------------------------------------------------------}
BEGIN { calculate }
   { Ground treated as specular. lmax is 2 for the reflection }
   IF albedo > 0.0 THEN lmax := 2 ELSE lmax := 1;

   { Original formulation used a different azimuth }
   tt := (theta + cangle) * degtorad; st := SIN(tt); ct := COS(tt);
   ee := elev * degtorad; se := SIN(ee); ce := COS(ee);

   {Initialize flux at detector}
   flu := 0.0;

   { Calculate flame geometry relative to detector's position and sort}
   sortbags;
   selector;

   FOR j := 1 TO nbags DO {for each flame}
     BEGIN { LOOP OVER ALL BAGS }
       xj := xbag[j]; yj := ybag[j]; zj := zbag[j];
       tj := tbag[j]; {not used yet}
```

```
rg2 := rg2bag[j]; rg := rgbag[j];
wl := wlbag[j]; wlr := wlrbag[j];

{Subdivide flame into m radiators}
mp := (safety * heightofbag) / rg + 1;
m := TRUNC(mp); IF m > mmax THEN m := mmax;
                IF m < mmin THEN m := mmin;

{ Note: zj is defined to be at the tip of the nozzle }

{Position each radiator along flame's axis of symmetry}
dz := heightofbag/m; zip[0] := -0.5 * dz + zj;
FOR i := 1 TO m DO zip[i] := zip[i-1] + dz;

{Calculate each radiator's power}
si := bagstrength;
sip[0] := 0.0; {using as the accumulator}
FOR i := 1 TO m DO {Normalize all radiators}
  BEGIN
    sip[i] := model( (zip[i] - zj)/heightofbag);
    sip[0] := sip[0] + sip[i];
  END;
FOR i := 1 TO m DO sip[i] := si * sip[i]/sip[0];

FOR i := 1 TO m DO {all radiators}
  BEGIN { loop over m points within flame }
```

```
zk := zip[i];
FOR l := 1 TO lmax DO {for direct and if lmax=2 then reflected}
  BEGIN { loop over points above and below ground }
    IF l > 1 THEN zk := -zk; { image of flame pt }
    r2 := rg2 + SQR(zdist - zk);
    r  := SQRT(r2); {range from source to detector}

    IF quick THEN {ignore obscuration}
      pbag[j] := sbag[j]
    ELSE {calculate obscuration} modbagstrength(j);

    si := sip[i] * pbag[j];
    nwlr := wlr; {use value from an outer loop}

    IF NOT quick THEN {approximate change in view angle}
      IF thavgi <> thavgf THEN {modify input to ca calculation}
        BEGIN
          te := (thavgf-theta) * degtorad; {a temporary}
          nwl := rg * (COS(te) * st + SIN(te) * ct);
          nwlr := nwl * ce;
        END;

    {ca is the cosine of the angle joining the detector and}
         {the source with the normal to the detector}
    ca := (nwlr - (zdist - zk) * se)/r;
    IF l > 1 THEN ca := ca * albedo; { ALBEDO IS 0-1 }
```

```
              delta_flux := si * ca / r2; {contribution from one source}
              IF ca > 0.0 THEN {it's visible} flu := flu + delta_flux;
            END;  { loop over above & below ground }
          END;  { loop over m points within bag }
        END;{ loop over bags }
      flu := flu * bestguessbag / fourpi; {calculate flux}
  END; { calculate }
{---------------------------------------------------------------------------}
BEGIN
  done := FALSE;
  default;
  REPEAT
    readdata; {read all the data in this routine. Sets the variable "done"}
    calculate; {calculate this case with this routine}
    WRITELN('Flux = ',flu); {print desired data here. This is an example}
  UNTIL done
END.
```

SECTION 7

RECOMMENDATIONS

This section presents our recommendations for enhancing the FC-TEE hardware. Based on a brief history of six months use on an Apple II+, SEA would recommend that DNA upgrade its systems with a Titan Technologies Accelerator. Unfortunately, the version for the Apple IIe is not yet available, and SEA must reserve such a recommendation until such time we have had an opportunity to test the new version. The advantage of such an upgrade would be to decrease the time it takes to perform any calculation. The Accelerator increases the speed of an Apple by about a factor of 3.5 for FC-TEE. A 3.6 MHz version of the 6502 is on board the Accelerator.

Another recommendation, also based on a limited use of six months, is the installation of a "RAM disk". This is a device that simulates a disk, but stores data in memory instead of diskettes. SEA recommends the PION, Inc. Interstellar Drive because it has battery backup and has its own power supply. Except for once a month maintenance, the memory will always be retained, unless power is removed for periods greater than 15 minutes.

APPENDIX A

LIST OF PHYSICS ROUTINES
WITH CROSS REFERENCE

```
>Pascal IDENTREF Lister:   OSI[6.1] for file >>> DOC-XLOX-F.TEXT <<<
{01 Nov 83 Documentation for PROGRAM trsradiation}
{$S+} {$V-}

PROGRAM trsradiation;

{----------------------------------------------------------------------------}
{   DOCUMENTATION of Algorithms & Models:  TRS Radiation Program - FLAMES  }
{----------------------------------------------------------------------------}
{ This version uses an inverted cone model for the TRS flames and accounts }
{ for flame obscuration, change in view angle.  It also uses a different   }
{ model for the height dependence of the flame's radiated power.           }
{----------------------------------------------------------------------------}
{                     AUTHOR: Burton S. Chambers, III                       }
{----------------------------------------------------------------------------}

USES
{$U *SYSTEM.LIBRARY}       {  applestuff, }
                              transcend,
{$U SYSLIB4:XTRS.LIBRARY}     mathtrap,
                              genmath;

CONST pi = 3.141592654; maxnobags = 50;

VAR
      fourpi, degtorad, radtodeg, albedo, hgtacoeff, hgtbcoeff: REAL;
```

```
    widthofbag, heightofbag, bestguessbag, bagstrength: REAL;
    xbag, ybag, zbag, tbag, sbag: ARRAY[1..maxnobags] OF REAL;
    nbags: INTEGER;

    quick: BOOLEAN;

    xdist, ydist, zdist, theta, elev, flu: REAL;
    done: BOOLEAN;
    wlbag, wlrbag, rg2bag, rgbag,
    pbag, abag, bbag: ARRAY[1..maxnobags] OF REAL;
    kbag: ARRAY[1..maxnobags] OF INTEGER;
{-----------------------------------------------------------------------}
  PROCEDURE readdata;

  BEGIN
{Eg.}
    xdist := 100.0;
    ydist :=   0.0;
    zdist := 600.0;
    theta :=   0.0;
    elev  :=   0.0;
    done := TRUE;
{Eg.}
  END;
{-----------------------------------------------------------------------}
  PROCEDURE default;
```

```
  VAR i,j:INTEGER;

  BEGIN
     fourpi := 4.0 * pi;
     degtorad := pi / 180; radtodeg := 1.0 / degtorad;
     bagstrength := 1.0; { RELATIVE TO BESTGUESSBAG }
     bestguessbag := 1.53372E7; { CAL/SEC }
     widthofbag := 274.32; { CM } {9'}
     heightofbag := 438.912; { CM } {14.4'}
     albedo := 0.2; { OF GROUND }
     hgtacoeff := 1.0;
     hgtbcoeff := 25.0;
     nbags := 1;
     FOR i := 1 TO maxnobags DO
       BEGIN { DEFAULT TO LINE OF EQUALLY SPACED NOZZLES }
         xbag[i] := 0.0; j := (i+1) DIV 2;
         ybag[i] := j * widthofbag - widthofbag/2;
         zbag[i] := 0.0;
         IF ODD(i) THEN ybag[i] := -ybag[i];
         tbag[i] := 0.0; sbag[i] := 1.0;
       END;
     quick := FALSE;
  END;
{-----------------------------------------------------------------------------}
 FUNCTION cintersection (xl, yl, rl, xn, yn, rn: REAL;
                        VAR d, xil, yil, xi2, yi2: REAL): BOOLEAN;
```

```
CONST pi = 3.14159265;
VAR alpha, theta: REAL;

BEGIN
  cintersection := FALSE;
  rl := ABS(rl); rn := ABS(rn); {make negative radii positive}
  d := sqrter(SQR(xn-xl)+SQR(yn-yl));
  IF (d > 0.0) AND (d <= rl + rn) THEN {at least one intersection exists}
    BEGIN
      theta := ataner(diver(yn-yl,xn-xl));
      IF xn < xl THEN {its in quadrant II or III} theta := theta + pi;
      alpha := acoser((SQR(rl) - SQR(d) - SQR(rn))/(2.0*d*rn));
      xi2 := xn + rn * coser(theta+alpha);
      yi2 := yn + rn * siner(theta+alpha);
      xil := xn + rn * coser(theta-alpha);
      yil := yn + rn * siner(theta-alpha);
      cintersection := TRUE;
    END;
END; {calcintersection}
{-----------------------------------------------------------------------------}
PROCEDURE calculate;

CONST cangle = 90.0; safety = 3.0;
      mmax = 51; { max set on basis of flame hgt to width data }
      mmin =  5; { min set on guess of flame obscuration effect}
```

```
VAR i,j,k,l,lmax,n,m:INTEGER;
    te,ee,se,ce,tt,st,ct:REAL;
    xj,yj,zj,sj,tj,rg,rg2,mp,si,wl,wlr,nwl,nwlr,zk,r,r2,ca:REAL;
    sip,zip:ARRAY[0..mmax] OF REAL;
    dz:REAL;
    thmax,thmin,thavgi,thavgf: REAL;
    delta_flux: REAL;
{---------------------------------------------------------------}
    FUNCTION model (z:REAL):REAL;
    VAR p: REAL;

    BEGIN
      IF z <= 1.0 THEN
        BEGIN
          IF hgtbcoeff = 1.0 THEN p := 1.0
          ELSE p := 1.0 + (1.0 - z) * (hgtbcoeff - 1.0)
        END
      ELSE IF z < hgtacoeff THEN
        BEGIN
          IF hgtacoeff <= 1.0 THEN p := 0.0 {shouldn't happen}
          ELSE p := 1.0 + (1.0 - z) / (hgtacoeff - 1.0)
        END
      ELSE p := 0;

      IF z < 1.0 THEN model := p * z * (widthofbag/2.0)
      ELSE model := p * (widthofbag/2.0);
```

```
  END; { model }
{-------------------------------------------------------------}
  PROCEDURE selector;
  TYPE  item = RECORD
               key: REAL;
               index: INTEGER;
             END;
  VAR kselect: ARRAY[1..maxnobags] OF item;
      i,j,k:INTEGER; x: item;

  BEGIN
    FOR j := 1 TO nbags DO kselect[j].key := rgbag[j];
    FOR j := 1 TO nbags DO kselect[j].index := j;

    FOR i := 1 TO nbags - 1 DO
      BEGIN
        k := i; x := kselect[i];
          FOR j := i + 1 TO nbags DO
            IF kselect[j].key < x.key THEN
              BEGIN
                k := j; x := kselect[j];
              END;
            kselect[k] := kselect[i]; kselect[i] := x;
      END;
    FOR j := 1 TO nbags DO kbag[j] := kselect[j].index;
  END;
```

```
{----------------------------------------------------------------}
   PROCEDURE sortbags;
   VAR w2,radius: REAL; j: INTEGER;

   BEGIN
     radius := widthofbag / 2.0;
     FOR j := 1 TO nbags DO
       BEGIN
         xj := xbag[j]; yj := ybag[j];
         rg2 := SQR(xdist-xj)+SQR(ydist-yj); rg := SQRT(rg2);
         w1 := (xdist - xj) * st + (ydist - yj) * ct;
         w2 := (ydist - yj) * st - (xdist - xj) * ct;
         w1r := w1 * ce;
         w1bag[j] := w1; w1rbag[j] := w1r;
         rg2bag[j] := rg2; rgbag[j] := rg;
         abag[j] := radtodeg * asiner(radius / rgbag[j]);
         { assumes quad I,IV, ie. abag in -90..+90, and not in flame }
         bbag[j] := radtodeg * acoser(w1 / rgbag[j]);
         { quad I,II, ie. bbag in 0..180 }
         IF w2 < 0.0 THEN { flame center to left of normal }
           bbag[j] := -bbag[j];
       END;
   END; { sortbags }
{----------------------------------------------------------------}
   PROCEDURE modbagstrength (l: INTEGER);
   { modify strength of one bag for this pt }
```

```
VAR j, k, m, n: INTEGER;
    ratio,  satio,  angle, anglemax, anglemin,
    zint,   ztip,   zsource: REAL;
    {dist = distance between circle-centers, xil,etc.=intersections}
    dist, xil,yil,xi2,yi2: REAL;
    radius,
    tang_dist: REAL;    {distance to tangency}
    intersect: BOOLEAN;
    sr, rr, deltaabag: REAL;

{---------------------------------------------------------}
  PROCEDURE switch (VAR x1,x2: REAL);
  VAR t: REAL;

  BEGIN
    t := x1; x1 := x2; x2 := t;
  END;
{---------------------------------------------------------}
  FUNCTION determinant (x1,y1,x2,y2,x3,y3: REAL): REAL;

  BEGIN
    determinant := x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2)
  END;
{---------------------------------------------------------}
  FUNCTION sameside: BOOLEAN;
  VAR pos1,pos2: BOOLEAN;
```

```
    BEGIN
      pos1 :=
        (determinant(xbag[1],ybag[1],xbag[m],ybag[m],xil,yil) > 0.0);
      pos2 :=
        (determinant(xbag[1],ybag[1],xbag[m],ybag[m],xdist,ydist) > 0.0);
      sameside := (pos1 AND pos2);
    END;
  {----------------------------------------------------------}
    PROCEDURE inmod;
    VAR inter1dist, inter2dist, beta1, beta2: REAL;

    {------------------------------------------------------}
      FUNCTION calcbeta (x,y,d: REAL): REAL;
      {Beta is relative to detector orientation}
      VAR w1,w2,beta: REAL;

      BEGIN
        w1 := (xdist - x) * st + (ydist - y) * ct;
        w2 := (ydist - y) * st - (xdist - x) * ct;
        beta := radtodeg * acoser(w1 / d);
        { quad I,II, ie. bbag in 0..180 }
        IF w2 < 0.0 THEN { intersection to left of normal }
          beta := -beta;
        calcbeta := beta;
      END;
```

```
{--------------------------------------------------------}
BEGIN
  tang_dist  := sqrter(rg2bag[m]-SQR(sr));
  inter1dist := sqrter(SQR(xdist-xi1)+SQR(ydist-yi1));
  inter2dist := sqrter(SQR(xdist-xi2)+SQR(ydist-yi2));
  beta1 := calcbeta(xi1,yi1,inter1dist);
  beta2 := calcbeta(xi2,yi2,inter2dist);
  IF inter1dist < tang_dist THEN {use intersection point 1}
    anglemin := bbag[m] - beta1;
  IF inter2dist < tang_dist THEN {use intersection point 2}
    anglemax := beta2 - bbag[m];
END;
{----------------------------------------------------------}
PROCEDURE modangles;

BEGIN
  angle := radtodeg * asiner(sr / rgbag[m]);
  anglemax := angle; anglemin := angle;
  IF intersect THEN inmod;
  IF thmin > bbag[m] - anglemin THEN
    IF bbag[m] + anglemax > thmin THEN
      thmin := bbag[m] + anglemax;
  IF thmax < bbag[m] + anglemax THEN
    IF bbag[m] - anglemin < thmax THEN
      thmax := bbag[m] - anglemin;
END;
```

```
{------------------------------------------------------------}
BEGIN
  radius := widthofbag / 2.0;
  j := 0;

  REPEAT {Find the bag index, j, so that kbag[j] points to 1}
    j := j + 1;
  UNTIL (1 = kbag[j]) OR (j = nbags);

  ratio := 1.0; ztip := zbag[1];
  zsource := ABS(zk);
  IF zsource < ztip THEN ratio := 0.0
  ELSE IF zsource < heightofbag + ztip THEN
    ratio := (zsource - ztip)/heightofbag;

  rr := ratio * radius; {for speed}
  deltaabag := radtodeg * asiner(rr / rgbag[1]);
  thmin  := bbag[1] - deltaabag;
  thmax  := bbag[1] + deltaabag;
  thavgi := bbag[1];           {0.5 * (thmin + thmax)}

  FOR n := 1 TO nbags DO {over all bags}
  {Those closer or intersecting with the source bag can obscure}
    BEGIN
      m := kbag[n];
      anglemin := 0.0; anglemax := 0.0; {for neat debug prints}
```

```
IF l <> m THEN {check for obscuration}
  BEGIN
    satio := 1.0; ztip := zbag[m];
    zint := ABS(zdist + (zk-zdist) * (rgbag[m] / rgbag[l]));

    IF zint < ztip THEN satio := 0.0
    ELSE IF zint < heightofbag + ztip THEN
      satio := (zint - ztip)/heightofbag;
    sr := satio * radius; {for speed}

    intersect := cintersection(
                    xbag[l],ybag[l],rr, {source}
                    xbag[m],ybag[m],sr, {obscuror}
                    dist,xil,yil,xi2,yi2);

    IF (intersect AND (NOT sameside)) THEN
                               {switch intersection points}
      BEGIN
        switch(xil,xi2); switch(yil,yi2);
      END;

    {Note: The next test must definitely be on n & j:  BSC3}
    IF n <= j THEN {the possibly obscuring flame is closer }
      modangles
    ELSE {it is farther away} IF intersect THEN modangles;

```

```
              END;
            END;
          thavgf := 0.5 * (thmin + thmax);
          IF (thmax > thmin) AND (ratio > 0.0) THEN
            pbag[l] := sbag[l] * (thmax - thmin) / (deltaabag + deltaabag)
          ELSE pbag[l] := 0.0;
        END; { modbagstrength }
    {-------------------------------------------------------------}
    BEGIN { calculate }
        IF albedo > 0.0 THEN lmax := 2 ELSE lmax := 1;
          { Ground treated as specular }
        tt := (theta + cangle) * degtorad; st := SIN(tt); ct := COS(tt);
        ee := elev * degtorad; se := SIN(ee); ce := COS(ee);
        flu := 0.0;

        sortbags; { position has been defined }
        selector;

        FOR j := 1 TO nbags DO
          BEGIN { LOOP OVER ALL BAGS }
            xj := xbag[j]; yj := ybag[j]; zj := zbag[j];
            tj := tbag[j];
            rg2 := rg2bag[j]; rg := rgbag[j];
            wl := wlbag[j]; wlr := wlrbag[j];

            mp := (safety * heightofbag) / rg + 1;
```

```
m := TRUNC(mp); IF m > mmax THEN m := mmax;
                IF m < mmin THEN m := mmin;

dz := heightofbag/m; zip[0] := -0.5 * dz + zj;
FOR i := 1 TO m DO zip[i] := zip[i-1] + dz;

si := bagstrength;
sip[0] := 0.0;
FOR i := 1 TO m DO {all source points}
  BEGIN
    sip[i] := model( (zip[i] - zj)/heightofbag);

    { zj is tip of nozzle }

    sip[0] := sip[0] + sip[i];
  END;
FOR i := 1 TO m DO sip[i] := si * sip[i]/sip[0];

FOR i := 1 TO m DO {all source points}
  BEGIN { loop over m points within flame }
    zk := zip[i];
    FOR l := 1 TO lmax DO
      BEGIN { loop over points above and below ground }
        IF l > 1 THEN zk := -zk; { image of flame pt }
        r2 := rg2 + SQR(zdist - zk);
        r  := SQRT(r2);
```

```
{QUICK-MOD}     IF quick THEN pbag[j] := sbag[j] ELSE modbagstrength(j);
                si := sip[i] * pbag[j];
                nwlr := wlr; {use value from an outer loop}
{QUICK-MOD}     IF NOT quick THEN
                  IF thavgi <> thavgf THEN {modify input to ca calculation}
                    BEGIN
                      te := (thavgf-theta) * degtorad; {a temporary}
                      nwl := rg * (COS(te) * st + SIN(te) * ct);
                      nwlr := nwl * ce;
                    END;
                ca := (nwlr - (zdist - zk) * se)/r;
                IF 1 > 1 THEN ca := ca * albedo; { ALBEDO IS 0-1 }
                delta_flux := si * ca / r2;
                IF ca > 0.0 THEN flu := flu + delta_flux;
              END;  { loop over above & below ground }
          END;  { loop over m points within bag }
      END;{ loop over bags }
    flu := flu * bestguessbag / fourpi;
  END; { calculate }
{--------------------------------------------------------------------------}
BEGIN
  done := FALSE;
  REPEAT
    readdata; {read all the data in this routine. Sets the variable "done"}
    calculate; {calculate this case with this routine}
    WRITELN('Flux = ',flu); {print desired data here}
```

```
  UNTIL done
END.
```

```
>Pascal IDENTREF List of Reserved words:   OSI[6.1]


AND          85   214   301   315
ARRAY        27*   35*   36*  107*  137*
BEGIN        40*   54*   66*   81*   86*  115*  117*  122*  140*  145*
            149*  160*  163*  196*  202*  209*  225*  235*  249*  261*
            283*  287*  303*  320*  331*  347*  357*  360*  369*  384*
CONST        22*   78*  100*
DIV          67
DO           65   141   142   144   147   154   162   281   330   342
            346   354   356   359
ELSE        119   121*  124   126*  129   272   292   310*  317   321
            364
END          49*   72*   74*   95*   96*  120   125   130*  136*  151*
            153*  155*  177*  178*  198*  204   215*  233*  245*  259*
            305*  312*  313*  318*  353*  373*  378*  379*  380*  382*
            391
FOR          65*  141*  142*  144*  147*  154*  162*  281*  330*  342*
            346*  354*  356*  359*
FUNCTION     76*  112*  200*  206*  221*
IF           70*   85*   88*  116*  118*  121*  123*  128*  148*  175*
            230*  241*  243*  252*  253*  254*  256*  257*  271*  272*
            286*  291*  292*  301*  308*  310*  315*  321*  338*  339*
            361*  364*  367*  368*  375*  377*
NOT         301   367
```

OF	27*	35*	36*	107*	137*					
OR	267									
PROCEDUR	38*	51*	98*	132*	157*	180*	193*	217*	247*	
PROGRAM	4*									
RECORD	133*									
REPEAT	265*	386*								
THEN	70	85	88	116	118	121	123	128	148	175
	230	241	243	252	253	254	256	257	271	272
	286	291	292	301	308	310	315	321	338	339
	361	364	367	368	375	377				
TO	65	141	142	144	147	154	162	281	330	342
	346	354	356	359						
TYPE	133*									
UNTIL	267*	390*								
USES	16*									
VAR	24*	52*	77*	79*	104*	113*	137*	158*	182*	193*
	194*	207*	218*	223*						

>Pascal IDENTREF List of Predefined words: OSI[6.1]

ABS	83	83	270	289						
BOOLEAN	30*	33*	77*	189*	206*	207*				
FALSE	73	82	385							
INTEGER	28*	36*	52*	104*	135*	138*	158*	180*	182*	
ODD	70									
REAL	25*	26*	27*	32*	35*	76*	77*	79*	105*	106*
	107*	108*	109*	110*	112*	112*	113*	134*	158*	184*
	186*	188*	190*	193*	194*	200*	200*	218*	221*	221*
	223*									
SQR	84	84	89	89	89	165	165	236	237	237
	238	238	362							
TRUE	47	94								
TRUNC	338									
WRITELN	389*									

>Pascal IDENTREF List of Transcend words: OSI[6.1]

```
COS       323   324   371
SIN       323   324   371
SQRT      165   363
```

```
>Pascal IDENTREF List of Identifiers:   OSI[6.1]


 abag        35*  171*
 acoser      89   173   228
 albedo      25*   61*  321   375
 alpha       79*   89*   90    91    92    93
 angle      183*  250*  251   251
 anglemax   183*  244*  251*  254   255   256   285*
 anglemin   183*  242*  251*  253   257   258   285*
 asiner     171   250   276
 ataner      87
 bagstren    26*   57*  344
 bbag        35*  173*  176*  176   242   244   253   254   255   256
            257   258   277   278   279
 bestgues    26*   58*  381
 beta       223*  228*  231*  231   232
 betal      218*  239*  242
 beta2      218*  240*  244
 ca         106*  374*  375*  375   376   377
 calcbeta   221*  232*  239   240
 calculat    98*  388*
 cangle     100*  323
 ce         105*  168   324*  372
 cinterse    76*   82*   94*  296
```

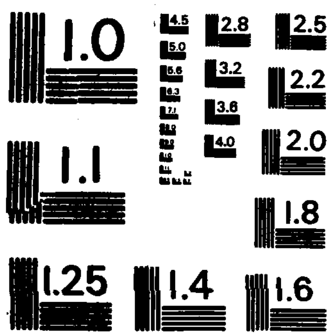
coser	90	92								
ct	105*	166	167	226	227	323*	371			
d	77*	84*	85	85	89	89	221*	228		
default	51*									
degtorad	25*	56*	56	323	324	370				
deltaaba	190*	276*	277	278	316	316				
deltaflu	110*	376*	377							
determin	200*	203*	211	213						
dist	186*	299								
diver	87									
done	33*	47*	385*	390						
dz	108*	341*	341	342						
ee	105*	324*	324	324						
elev	32*	46*	324							
flu	32*	325*	377*	377	381*	381	389*			
fourpi	25*	55*	381							
genmath	20*									
heightof	26*	60*	272	273	292	293	337	341	348	
hgtacoef	25*	62*	121	123	124					
hgtbcoef	25*	63*	118	119						
i	52*	65*	67*	67	68*	69*	70	70*	70	71*
	71*	104*	138*	144*	146	146	147	152	152*	342*
	342*	342	346*	348*	348	352	354*	354*	354	356*
	358	365								
index	135*	142*	154							
inmod	217*	252*								

inter1di	218*	237*	239	241						
inter2di	218*	238*	240	243						
intersec	189*	252	296*	301	310					
item	133*	137*	138*							
j	52*	67*	68	104*	138*	141*	141*	141	142*	142*
	142	147*	148	150	150	154*	154*	154	158*	162*
	164	164	169*	169*	170*	170*	171*	171	173*	173
	176*	176	182*	263*	266*	266	267	267	308	330*
	332	332	332	333	334	334	335	335	364*	364
	364*	365								
k	104*	138*	146*	150*	152*	182*				
kbag	36*	154*	267	284						
key	134*	141*	148	148						
kselect	137*	141*	142*	146	148	150	152*	152	152*	154
l	104*	180*	211	211	213	213	267	269	276	277
	278	279	286	289	297	297	316*	316	317*	359*
	361	375								
lmax	104*	321*	321*	359						
m	104*	182*	211	211	213	213	236	242	244	250
	253	254	255	256	257	258	284*	286	288	289
	298	298	338*	338	338*	339	339*	341	342	346
	354	356								
mathtrap	19*									
maxnobag	22*	27*	35*	36*	65	137*				
mmax	101*	107*	338	338						
mmin	102*	339	339							

modangle	247*	309*	310*							
modbagst	180*	364*								
model	112*	128*	129*	348						
mp	106*	337*	338							
n	104*	182*	281*	284	308					
nbags	28*	64*	141	142	144	147	154	162	267	281
	330									
nwl	106*	371*	372							
nwlr	106*	366*	372*	374						
p	113*	118*	119*	123*	124*	126*	128	129		
pbag	35*	316*	317*	364*	365					
pi	22*	55	56	78*	88					
pos1	207*	210*	214							
pos2	207*	212*	214							
quick	30*	73*	364	367						
r	106*	363*	374							
r2	106*	362*	363	376						
radius	158*	161*	171	187*	262*	275	294			
radtodeg	25*	56*	171	173	228	250	276			
ratio	183*	269*	271*	273*	275	315				
readdata	38*	387*								
rg	106*	165*	170	334*	337	371				
rg2	106*	165*	165	170	334*	362				
rg2bag	34*	170*	236	334						
rgbag	34*	141	170*	171	173	250	276	289	289	334
r1	76*	83*	83	85	89					

AD-A166 108 IMPLEMENTATION OF THERMAL MODELS AND ALGORITHMS AT FIELD COMMAND DNA (DFF (U) SCIENCE AND ENGINEERING ASSOCIATES INC SALEM MA B S CHAMBERS 12 DEC 84 2/2

UNCLASSIFIED SEA-84-4 DNA-TR-85-4 DNA001-83-C-0257 F/G 18/8 NL


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

rn	76*	83*	83	85	89	89	90	91	92	93
rr	190*	275*	276	297						
safety	100*	337								
sameside	206*	214*	301							
satio	183*	288*	291*	293*	294					
sbag	27*	71*	316	364						
se	105*	324*	374							
selector	132*	328*								
si	106*	344*	354	365*	376					
siner	91	93								
sip	107*	345*	348*	352*	352	352	354*	354	354	365
sj	106*									
sortbags	157*	327*								
sqrter	84	236	237	238						
sr	190*	236	250	294*	298					
st	105*	166	167	226	227	323*	371			
switch	193*	304*	304*							
t	194*	197*	197							
tangdist	188*	236*	241	243						
tbag	27*	71*	333							
te	105*	370*	371	371						
thavgf	109*	314*	368	370						
thavgi	109*	279*	368							
theta	32*	45*	79*	87*	88*	88	90	91	92	93
	323	370								
thmax	109*	256	257	258*	278*	314	315	316		

thmin	109*	253	254	255*	277*	314	315	316		
tj	106*	333*								
transcen	18*									
trsradia	4*									
tt	105*	323*	323	323						
wl	106*	166*	168	169	173	223*	226*	228	335*	
wlbag	34*	169*	335							
wlr	106*	168*	169	335*	366					
wlrbag	34*	169*	335							
w2	158*	167*	175	223*	227*	230				
widthofb	26*	59*	68	68	128	129	161	262		
x	138*	146*	148	150*	152	221*	226	227		
xl	193*	197	197*	200*	203					
x2	193*	197	197*	200*	203					
x3	200*	203								
xbag	27*	67*	164	211	211	213	213	297	298	332
xdist	32*	42*	165	166	167	213	226	227	237	238
xil	77*	92*	186*	211	237	239	299	304*		
xi2	77*	90*	186*	238	240	299	304*			
xj	106*	164*	165	166	167	332*				
xl	76*	84	87	88						
xn	76*	84	87	88	90	92				
y	221*	226	227							
yl	200*	203	203							
y2	200*	203	203							
y3	200*	203	203							

ybag	27*	68*	70*	70	164	211	211	213	213	297
	298	332								
ydist	32*	43*	165	166	167	213	226	227	237	238
yi1	77*	93*	186*	211	237	239	299	304*		
yi2	77*	91*	186*	238	240	299	304*			
yj	106*	164*	165	166	167	332*				
yl	76*	84	87							
yn	76*	84	87	91	93					
z	112*	116	119	121	124	128	128			
zbag	27*	69*	269	288	332					
zdist	32*	44*	289	289	362	374				
zint	184*	289*	291	292	293					
zip	107*	341*	342*	342	348	358				
zj	106*	332*	341	348						
zk	106*	270	289	358*	361*	361	362	374		
zsource	184*	270*	271	272	273					
ztip	184*	269*	271	272	273	288*	291	292	293	

APPENDIX B

LIST OF IDENTIFIERS
AND THEIR MEANINGS

FCTEE Identifiers and their meanings.

Each identifier is truncated to the first eight characters, exclusive of the underscore character, since the version of Pascal used for FCTEE does not recognize additional characters. Any variable used as a simple index has been excised from this listing.

abag .... Half angle view of jth flame at field point.

acoser ...... Special Function arc cosine

albedo ...... Ground albedo

alpha .... In cintersection it is the complement of the angle that is formed by the line of centers with the normal to the line of centers from the intersection point.

angle .... Temporary used in computing angle extents of current flame that is visible to the field point.

anglemax .... Maximum angle to one extremum of visible edge of current flame as seen by field point.

anglemin .... Minimum angle to one extremum of visible edge of current flame as seen by field point.

asiner ...... Special Function arc sine

ataner ...... Special Function arc tangent

bagstren .... Model parameter relative flame power.

bbag .... Angle in xy plane to center of jth flame at field point relative to the field point's normal.

bestgues .... Model parameter best guess of flame power.

beta .... Bbag for jth flame.

betal .... Beta to intersection point 1 from field point.

beta2 .... Beta to intersection point 2 from field point.

ca .... Cosine of the angle between the line segment, that joins the field point and the current source point, and the normal to the field point's surface.

calcbeta .... Procedure that is part of algorithm to modify angles. See inmod.

calculat .... Procedure to calculate modeled flux at one field point.

cangle ...... Constant. Angle to add to original definition of theta to obtain azimuth. User inputs azimuth where zero azimuth is along x-axis and points towards negative x. Positive azimuth is towards positive y-axis.

ce .... Cosine of ee.

cinterse .... Procedure to calculate circle intersections with other circles.

coser .... Special Function cosine

ct .... Cosine of tt.

d .... Distance. Eg. Output from cintersection. distance between centers of the two circles. In calcbeta it is an argument where the distance between the field point and the point at which the angle beta is to be calculated is stored. Beta is relative to the field point's azimuth.

default ..... Procedure to provide default values for globals.

degtorad .... Conversion factor (degrees to radians)

deltaaba .... Angular width of source flame.

deltaflu .... The amount of flux to be added by this one source element.

determin .... Procedure to compute the value of determinant for evaluating which side point 3 is on compared to line from points 1 to 2.

dist .... Store output distance between circle centers from cintersection. Not used.

diver .... Special Function divide

done .... If true then user is finished doing calculations (done).

dz .... Space between source elements on flame.

ee .... Elevation of normal to field point in radians.

elev .... Elevation of normal to field point in degrees.

flu .... Variable to store calculated flux at field point.

fourpi ...... Four times pi

heightof .... Model parameter height of flame.

hgtacoef .... Model parameter height_a_coefficient.

hgtbcoef .... Model parameter height_b_coefficient.

index .... Field for record type item. quantity to be store pointer that

shows order of sorted ground ranges in procedure selector.

inmod .... Procedure that is part of algorithm to modify angles. See modangles.

inter1di .... Ground range to intersection point 1 from field point.

inter2di .... Ground range to intersection point 2 from field point.

intersec .... If true then two circles (flames) intersect.

item .... Record structure for sorting.

kbag .... Jth pointer. points to flame 1.

key .... Field for record type item. quantity to be sorted upon is ground range from field point to flame center.

kselect ..... Array of record type item to store data to be sorted by procedure selector.

lmax .... Number of times to add source element's contribution. It is set equal to one if albedo of ground is zero, otherwise set to two, since the source element and its image on the ground contribute.

maxnobag .... Constant. Maximum number of flames that can be considered.

mmax .... Constant. Maximum number of radiating points along any flame to obtain acceptable accuracy.

mmin .... Constant. Minimum number of radiating points along any flame to obtain acceptable accuracy.

modangle .... Procedure to modify angles for obscuring flames.

modbagst .... Procedure to modify the radiating strength for one flame that arrives at the current field point.

model .... Function to calculate normalized power per unit height multiplied by the width of the flame for this source point.

mp .... Number of source elements on flame.

nbags .... Input data. Number of flames.

nwl .... New value of wl. Modify if obscuration effect.

nwlr .... New value of wlr. Modify if obscuration effect.

p .... Temporary

pbag .... Calculated relative power of ith flame. This is set equal to sbag, unless obscuration exists.

pi 3.141592654

pos1 .... Temporary. If true then point 1 on positive side.

pos2 .... Temporary. If true then point 2 on positive side.

quick .... If true then skip flame-flame obscuration effects.

r .... Range from source point to field point.

r2 .... r squared.

radius ...... Temporary for radius of flame at height of flame.

radtodeg .... Conversion factor (radians to degrees)

ratio .... Ratio of height of source above nozzle to its total flame height.

readdata .... Procedure to read input data.

rg .... Rgbag for jth flame.

rg2 .... Rg2bag for jth flame.

rg2bag ...... Rgbag squared.

rgbag .... Ground range from field point to jth flame center.

rl .... Input to cintersection. radius of the lth circle.

rn .... Input to cintersection. radius of the nth circle.

rr .... Radius of source flame at source point height.

safety ...... Set to obtain conservative criterion for accuracy.

sameside .... Procedure to determine whether two special points are on the same side of the line joining the lth and mth flames.

satio .... Ratio of height of obscuror above nozzle to its total flame height.

sbag .... Input data. strength of ith flame.

se .... Sine of ee.

selector .... Procedure to sort flames relative to distance from field point.

si .... Initially a temporary for source strength for entire flame, but then modified to be source strength for ith source element, which is why it was named the way it was.

siner .... Special Function sine

sip .... Normalized power per unit height multiplied by the width of the flame for ith source point, which is in turn normalized by all the other sources in this flame.

sj .... Sbag for jth flame.

sortbags .... Procedure to calculate parameters in preparation for the sort that Procedure selector will perform.

sqrter ...... Special Function square root.

sr .... Radius of obscuror flame at relevant height. The relevant height depends on where the field point is and where the source flame is. The height is that where the line joining these two points cuts through the obscuring flame.

st .... Sine of tt.

switch ...... Procedure to switch the values of two real variables.

t .... Temporary

tangdist .... Ground range to point of tangency on flame from field point.

tbag .... Not used.

te .... Temporary.

thavgf ...... Final average of thmin and thmax.

thavgi ...... Initial average of thmin and thmax.

theta .... Azimuth heading of normal to field point. In cintersection it is angle that line of centers makes with y-axis.

thmax .... Maximum azimuth corresponding to mth flame's beta minus its anglemin.

thmin .... Minimum azimuth corresponding to mth flame's beta plus its anglemax.

tj .... Tbag for jth flame.

trsradia .... Program name

tt .... Original formulation used this angle instead of azimuth.

wl .... Wlbag for jth flame.

wlbag .... Cosine beta for field point and jth flame.

wlr .... Wlrbag for jth flame.

wlrbag ...... Projection of cosine beta on ground for field point and jth flame.

w2 .... Sine beta for field point and jth flame. used to determine whether flame center is to left of normal, where the vector direction is important to definition of left.

widthofb .... Model parameter width of flame.

x .... In selector it is a temporary. In calcbeta it is an argument where the x-coordinate is stored for the point at which the angle beta is to be calculated. Beta is relative to the field point's azimuth.

x1 .... Argument. Any real value or x for point 1

x2 .... Argument. Any real value or x for point 2

x3 .... Argument. x for point 3

xbag .... Input data. x-position of ith flame.

xdist .... X-position of field point.

xil .... Output from cintersection. x-coordinate of 1st intersection point.

xi2 .... Output from cintersection. x-coordinate of 2nd intersection point.

xj .... Xbag for jth flame.

xl .... Input to cintersection. x-coordinate of center of the lth circle.

xn .... Input to cintersection. x-coordinate of center of the nth circle.

y .... In calcbeta it is an argument where the y-coordinate is stored for the point at which the angle beta is to be calculated. Beta is relative to the field point's azimuth.

y1 .... Argument. y for point 1

y2 .... Argument. y for point 2

y3 .... Argument. y for point 3

ybag .... Input data. y-position of ith flame.

ydist .... Y-position of field point.

yi1 .... Output from cintersection. y-coordinate of 1st intersection

point.

yi2 .... Output from cintersection. y-coordinate of 2nd intersection point.

yj .... Ybag for jth flame.

yl .... Input to cintersection. y-coordinate of center of the lth circle.

yn .... Input to cintersection. y-coordinate of center of the nth circle.

z .... In model this is height normalized by flame height.

zbag .... Input data. z-position of ith flame.

zdist .... Z-position of field point.

zint .... Intersection height of obscuror above its nozzle height.

zip .... Height of the flame for ith source point.

zj .... Zbag for jth flame.

zk .... Zip for ith source point.

zsource ..... Height of source above its nozzle height.

ztip .... Height of bottom of flame (its tip).

APPENDIX C

DEFINITION OF ANGLE - VARIABLES

The next two figures define some of the angles described in Appendices A and B. They are included for use by any user wishing to better understand the basis for the equations found in Appendix A.

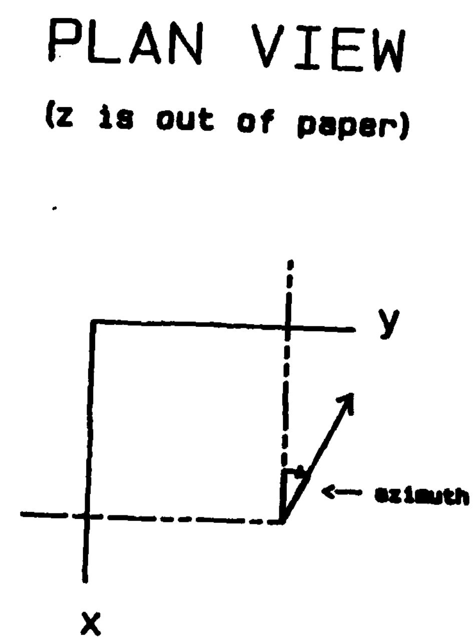


Figure 17. Definition of azimuth.

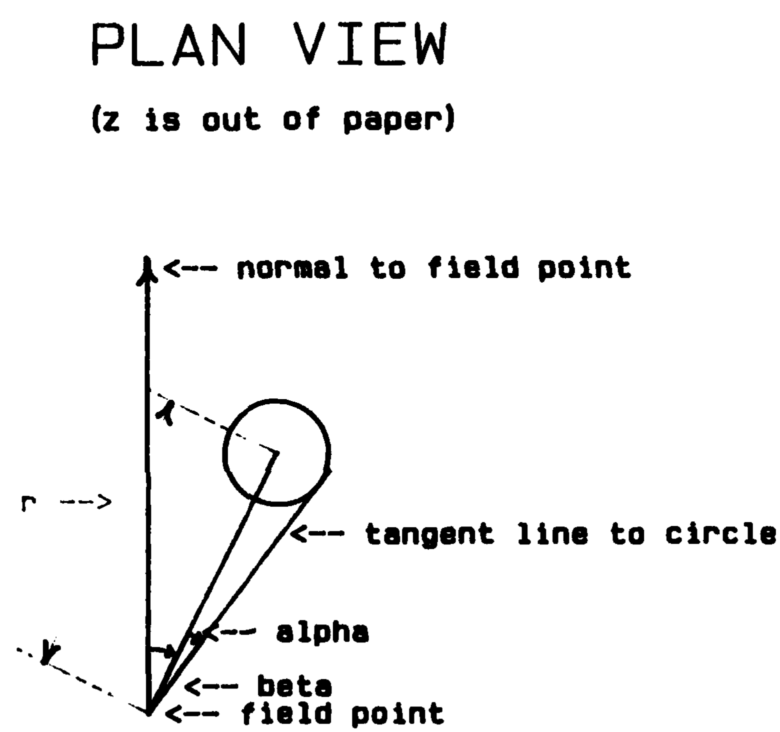


Figure 18. Definition of alpha and beta.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Defense Intell Agency
ATTN: RTS-2A, Tech Lib
ATTN: RTS-2B

Defense Nuclear Agency
ATTN: SPTD
4 cys ATTN: STTI-CA

Defense Tech Info Center
12 cys ATTN: DD

Field Command, Defense Nuclear Agency
ATTN: FCPR
ATTN: FCT
ATTN: FCTEI
ATTN: FCTT
ATTN: FCTT, W. Summa
ATTN: FCTXE

Under Secy of Def for Rsch & Engrg
ATTN: Engr Tech, J. Persh

DEPARTMENT OF THE ARMY

BMD Advanced Technology Ctr
ATTN: ATC-O, F. Hoke
ATTN: BMDATC-D, M. Capps
ATTN: 1CRDABH-X

BMD Program Office
ATTN: DACS-BMT
ATTN: DACS-BMZ

BMD Systems Command
ATTN: BMDSC-HW

Harry Diamond Laboratories
ATTN: DELHD-DTSO, 00103
ATTN: DELHD-NW-RA, L. Belliveau, 22100
ATTN: DELHD-TA-L, 81100, Tech Lib
ATTN: SLCHD-NW-P

US Army Ballistic Research Lab
ATTN: DRDAR-BLA-S, Tech Lib
ATTN: DRDAR-BLT

US Army Engr Waterways Exper Station
ATTN: Library
ATTN: WESSE

US Army Foreign Science & Tech Ctr
ATTN: DRXST-SD

US Army Mobility Equip R&D Cmd
ATTN: DRDME-WC, Tech Lib

US Army Nuc & Chem Agency
ATTN: Library

US Army Tank Automotive R&D Command
ATTN: DRDTA-UL, Tech Lib

US Army White Sands Missile Range
ATTN: STEWS-TE-N

DEPARTMENT OF THE ARMY (Continued)

USA Missile Command
ATTN: DRSMI

DEPARTMENT OF THE NAVY

David Taylor Naval Ship R&D Ctr
ATTN: Tech Info Ctr, Code 522.1

Naval Research Laboratory
ATTN: Code 2627, Tech Lib

Naval Surface Weapons Center
ATTN: Code X211, Tech Lib

Naval Surface Weapons Center
ATTN: Tech Lib & Info Svcs Br

Naval Weapons Center
ATTN: Code 343, FKA6A2, Tech Svcs

Naval Weapons Evaluation Facility
ATTN: Code 10, Tech Lib

Space & Naval Warfare Systems Cmd
ATTN: PME 117-21

Strategic Systems Programs, PM-1
ATTN: NSP-43, Tech Lib

DEPARTMENT OF THE AIR FORCE

Aeronautical Systems Div
ATTN: ASD/ENSSA

Air Force Inst of Tech, Air University
ATTN: Library

Air Force Weapons Laboratory
ATTN: NTE

Foreign Technology Division
ATTN: NIIS, Library

Strategic Air Command
ATTN: DOCSD
ATTN: DOWE
ATTN: INAO
ATTN: NRI/STINFO

DEPARTMENT OF ENERGY

Dept of Energy, Albuquerque Opns Office
ATTN: Tech Lib

University of California
Lawrence Livermore National Lab
ATTN: Tech Info Dept Lib

Los Alamos National Laboratory
ATTN: MS P364, Reports Library

Sandia National Laboratories
ATTN: Tech Lib, 3141

DEPARTMENT OF DEFENSE CONTRACTORS

California Research & Technology, Inc
ATTN: F. Sauer

Kaman Tempo
ATTN: DASIAC

Science & Engrg Associates, Inc
2 cys ATTN: B. Chambers, III

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Kaman Tempo
ATTN: DASIAC
ATTN: J. Shoutens
ATTN: W. Chan

Science Applications International Corp
ATTN: G. Binninger

END

DTIC

5-86